

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing existing information, gathering the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December, 1995	3. REPORT Final	0816
4. TITLE AND SUBTITLE USAF Summer Research Program - 1995 Graduate Student Research Program Final Reports, Volume 11, AEDC and WHMC		5. FUNDING NUMBERS	
6. AUTHORS Gary Moore			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Research and Development Labs, Culver City, CA		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NI 4040 Fairfax Dr, Suite 500 Arlington, VA 22203-1613		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Contract Number: F49620-93-C-0063			
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The United States Air Force Graduate Student Research Program (USAF- GSRP) is designed to introduce university, college, and technical institute graduate students to Air Force research. This is accomplished by the graduate students being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory Technical Directorates and Air Force Air Logistics Centers. Each participant provided a report of their research, and these reports are consolidated into this annual report.			
14. SUBJECT TERMS AIR FORCE RESEARCH, AIR FORCE, ENGINEERING, LABORATORIES, REPORTS, SUMMER, UNIVERSITIES		15. NUMBER OF PAGES	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

UNITED STATES AIR FORCE
SUMMER RESEARCH PROGRAM -- 1995
GRADUATE STUDENT RESEARCH PROGRAM FINAL REPORTS

VOLUME 11

ARNOLD ENGINEERING DEVELOPMENT CENTER
WILFORD HALL MEDICAL CENTER

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way
Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major David Hart

Program Manager, RDL
Scott Licoscas

Program Administrator, RDL
Gwendolyn Smith

Program Administrator, RDL
Johnetta Thompson

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1995

19981214 078

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

This document is one of a set of 16 volumes describing the 1995 AFOSR Summer Research Program. The following volumes comprise the set:

VOLUME

TITLE

1. Program Management Report

Summer Faculty Research Program (SFRP) Reports

2A & 2B Armstrong Laboratory

3A & 3B Phillips Laboratory

4 Rome Laboratory

5A, 5B, & 5C Wright Laboratory

6A & 6B Arnold Engineering Development Center, Wilford Hall Medical Center and
Air Logistics Centers

Graduate Student Research Program (GSRP) Reports

7A & 7B Armstrong Laboratory

8 Phillips Laboratory

9 Rome Laboratory

10A & 10B Wright Laboratory

11 Arnold Engineering Development Center, Wilford Hall Medical Center and
Air Logistics Centers

High School Apprenticeship Program (HSAP) Reports

12A & 12B Armstrong Laboratory

13 Phillips Laboratory

14 Rome Laboratory

15A&15B Wright Laboratory

16 Arnold Engineering Development Center

GSRP FINAL REPORT TABLE OF CONTENTS

i-xiv

1. INTRODUCTION	1
2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM	2
3. RECRUITING AND SELECTION	3
4. SITE VISITS	4
5. HBCU/MI PARTICIPATION	4
6. SRP FUNDING SOURCES	5
7. COMPENSATION FOR PARTICIPATIONS	5
8. CONTENTS OF THE 1995 REPORT	6

APPENDICIES:

A. PROGRAM STATISTICAL SUMMARY	A-1
B. SRP EVALUATION RESPONSES	B-1

GSRP FINAL REPORTS

A Methodology for Assessing Experimental Uncertainties in Curvefits
with Application to Compressor Map Characteristics

Kendall K. Brown
Graduate Research Assistant
Department of Mechanical and Aerospace Engineering

University of Alabama in Huntsville
RI Building E-13
Propulsion Research Center
Huntsville, AL 35899

Final Report for:
Graduate Student Research Program
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and
Arnold Engineering Development Center

August 1995

A Methodology for Assessing Experimental Uncertainties in Curvefits
with Application to Compressor Map Characteristics

Kendall K. Brown
Graduate Research Assistant
Department of Mechanical and Aerospace Engineering
University of Alabama in Huntsville

Abstract

In this report a methodology is presented to assess the uncertainties associated with curvefits and is demonstrated with an application to turbine engine compressor map characteristics.. The approach presented is a continuation of work to develop a methodology to properly account for these uncertainties. It is based upon applying uncertainty propagation techniques to the linear regression analysis equations used for curvefits. It assesses the uncertainty when precision, systematic, and correlated systematic uncertainties are present. This technique can also be applied when the X and Y variables are functional relations of test data, often referred to as data reduction equations. The methodology is presented for first order curvefits and is extended to the general n^{th} order curvefit. The effectiveness of this methodology was analyzed using Monte Carlo-type simulations. The conclusion of this report is that this methodology for assessing the uncertainty in curvefits is acceptable for 1st order linear regressions. The extension of the methodology to n^{th} order polynomial curvefits requires further research.

A Methodology for Assessing Experimental Uncertainties in Curvefits with Application to Compressor Map Characteristics

Kendall K. Brown

Introduction

When experimental information is used in a model in the form of a curvefit an uncertainty due to the original experimental program is introduced. Currently there is no accepted methodology to assess this uncertainty. The work reported here is a continuation of a previous effort¹ to develop a methodology to assess the uncertainty in linear regression analysis, or curvefits, when the experimental uncertainties contain precision, systematic, and correlated systematic uncertainties. The work reported in reference [1] developed a methodology to assess the uncertainty in the regression coefficients, slope and y-intercept, for a 1st order linear regression. The objective of this research is to extend this methodology to regression models used in turbine engine modeling at Arnold Engineering Development Center (AEDC). The most common forms of regression models used are straight lines and polynomial curvefits where both the X and Y axes are functional relationships containing uncertain experimental information. While multivariate regression models are also important models, they were not studied in this effort.

Linear regression analysis equations are based upon some fundamental statistical assumptions. When applying these equations to typical engineering data many of these underlying assumptions are violated. The underlying assumptions assume that the errors “(1) are unbiased; (2) have constant variance; (3) are uncorrelated, and (4) are normally distributed.”² In a typical experimental program the first three assumptions will usually be violated. Seber² and Montgomery³ discuss these assumptions and provide methods to deal with some of the violations, however their methods are mathematically rigorous and do not readily lend themselves to the propagation of experimental uncertainties. Violation of these assumptions does not preclude using the regression models to obtain a model for the data, but the model must be used with the realization that it only represents the “best fit” of a curve through the data. Any confidence interval then used with that curvefit must properly account for any violation of the underlying assumptions.

The methodology presented in this report is an extension of accepted uncertainty propagation techniques^{4,5,6,7,8} to the linear regression equations. The effectiveness of this new methodology was evaluated using Monte Carlo-type simulations. This report will briefly discuss the uncertainty analysis techniques and then present its application to linear regression analysis, first for 1st order (straight line) regressions and then for higher order (polynomial) regressions followed by a summary of the Monte Carlo simulation technique employed to evaluate the methodology. The report

concludes with the application of the curvefit uncertainty methodology to compressor map characteristics

Uncertainty Analysis Background

A brief overview of the methodology to obtain uncertainty estimates and how they propagate through a given data reduction equation is given in this report. The reader is referred to references [4] through [8] for an in-depth discussion.

The word accuracy is generally used to indicate the relative closeness of agreement between an experimentally-determined value of a quantity and its true value. Error is the difference between the experimentally-determined value and the truth, thus as error decreases accuracy is said to increase. Only in rare instances is the true value of a quantity known. Thus, it is necessary to estimate error, and that estimate is called an uncertainty, U . Uncertainty estimates are made at some confidence level -- a 95% confidence estimate, for example, means that the true value of the quantity is expected to be within the $\pm U$ interval about the experimentally-determined value 95 times out of 100.

Total error can be considered to be composed of two components: a precision (random) component ϵ and a bias (systematic) component β . An error is classified as precision if it contributes to the scatter of the data; otherwise, it is a bias error. As an estimator of β , a systematic uncertainty or bias limit, B , is defined. A useful approach to estimating the magnitude of a bias error is to assume that the bias error for a given case is a single realization drawn from some statistical parent distribution of possible bias errors. As an estimator of the magnitude of the precision errors, a precision uncertainty or precision limit P is defined

In nearly all experiments, the measured values of different variables are combined using a data reduction equation (DRE) to form some desired result. A general representation of a data reduction equation is

$$r = r(X_1, X_2, \dots, X_J) \quad (1)$$

where r is the experimental result determined from J measured variables X_i . Each of the measured variables contains bias errors and precision errors. These errors in the measured values then propagate through the data reduction equation, thereby generating the bias and precision errors in the experimental result, r .

If the "large sample assumption"⁶ is made, then the 95% confidence expression for U_r becomes

$$U_r^2 = B_r^2 + P_r^2 \quad (2)$$

where we define the systematic uncertainty (bias limit) of the result as

$$B_r^2 = \sum_{i=1}^J \theta_i^2 B_i^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \theta_i \theta_k B_{ik} \quad (3)$$

and the precision uncertainty (precision limit) of the result as

$$P_r^2 = \sum_{i=1}^J \theta_i^2 P_i^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \theta_i \theta_k P_{ik} \quad (4)$$

and where B_{ik} is the 95% confidence estimate of the covariance appropriate for the bias errors in X_i and X_k and P_{ik} is the 95% confidence estimate of the covariance appropriate for the precision errors in X_i and X_k , and the partial derivatives are

$$\theta_i = \frac{\partial r}{\partial X_i} \quad (5)$$

The 95% confidence precision limit for a variable X_i is estimated as

$$P_{X_i} = 2S_{X_i} \quad (6)$$

where the sample standard deviation for X_i is

$$S_{X_i} = \left[\frac{1}{N-1} \sum_{k=1}^N [(X_i)_k - \bar{X}_i]^2 \right]^{1/2} \quad (7)$$

and the mean value for X_i is defined as

$$\bar{X}_i = \frac{1}{N} \left[\sum_{k=1}^N (X_i)_k \right] \quad (8)$$

Typically, correlated precision uncertainties have been neglected so that the P_{ik} 's in Eq. (4) are taken as zero, and that is assumed in the work reported here. For a thorough discussion of the estimation of the B_{ik} 's in Eq. (3), the reader is referred to Reference [8].

Regression Analysis Uncertainty Methodology for Slope and y-Intercept

The uncertainties in the slope, m , and the Y-intercept, c , are functions of the uncertainties in the determinations of the X and Y variables. The values of m and c are obtained by minimizing the sum of the squares of the deviations between the line and the data points, commonly known as the method of least squares. The development of the equations to calculate m and c can be found in statistics books^{2,3,9} and only the equations will be presented here. For N (X_i , Y_i) data pairs, the slope of the line, m , is determined from

$$m = \frac{N \sum_{i=1}^N X_i Y_i - \sum_{i=1}^N X_i \sum_{i=1}^N Y_i}{N \sum_{i=1}^N (X_i^2) - \left(\sum_{i=1}^N X_i \right)^2} \quad (9)$$

and the intercept, c , is determined from

$$c = \frac{\sum_{i=1}^N (X_i^2) \sum_{i=1}^N Y_i - \sum_{i=1}^N X_i \sum_{i=1}^N (X_i Y_i)}{N \sum_{i=1}^N (X_i^2) - \left(\sum_{i=1}^N X_i \right)^2} \quad (10)$$

Considering Eq.s (9) and (10) to be data reduction equations of the form

$$m = m(X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_N) \quad (11)$$

and

$$c = c(X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_N) \quad (12)$$

and applying the uncertainty analysis equations, Eq.s (1)-(8), the most general form of the expression for the uncertainty in the slope of the line, m , is

$$\begin{aligned} U_m^2 = & \sum_{i=1}^J \left(\frac{\partial m}{\partial Y_i} \right)^2 B_{Y_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial m}{\partial Y_i} \right) \left(\frac{\partial m}{\partial Y_k} \right) B_{Y_i Y_k} + \sum_{i=1}^J \left(\frac{\partial m}{\partial Y_i} \right)^2 P_{Y_i}^2 \\ & + \sum_{i=1}^J \left(\frac{\partial m}{\partial X_i} \right)^2 B_{X_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial m}{\partial X_i} \right) \left(\frac{\partial m}{\partial X_k} \right) B_{X_i X_k} + \sum_{i=1}^J \left(\frac{\partial m}{\partial X_i} \right)^2 P_{X_i}^2 \\ & + 2 \sum_{i=1}^J \sum_{k=1}^J \left(\frac{\partial m}{\partial X_i} \right) \left(\frac{\partial m}{\partial Y_k} \right) B_{X_i Y_k} \end{aligned} \quad (13)$$

where B_{Y_i} is the systematic uncertainty for the Y_i variable, B_{X_i} is the systematic uncertainty for the X_i variable, $B_{Y_i Y_k}$ is the covariance estimator for the correlated bias uncertainties in the Y_i and Y_k variables, $B_{X_i X_k}$ is the covariance estimator for correlated bias uncertainties in the X_i and X_k variables, P_{Y_i} is the random uncertainty for the Y_i variable and P_{X_i} is the random uncertainty for the X_i variable, and $B_{X_i Y_k}$ is the covariance estimator for the correlated bias uncertainties between X_i and Y_k .

A similar expression for the uncertainty in the y-intercept is easily obtained by replacing the m 's in Eq. (13) with c 's. Analytical expressions for the partial derivatives are given in reference [1], or they can be determined numerically, as discussed later.

Equation (13) above shows the most general form of the equation for the uncertainty in the slope, and allows for correlation of bias errors among the different X 's, among the different Y 's, and also among the X 's and Y 's. If none of the systematic error sources are common between the X variables and the Y variables, the last term of the equations, the X - Y covariance estimator, is zero.

Regression Analysis Methodology for Predicted Value: Straight Line

The expression for the uncertainty in the predicted value, Y , at a given X variable

$$Y(X) = mX + c \quad (14)$$

(where X is assumed to have no uncertainty) is determined in a similar manner by applying Eq.s (2) through (8) to Eq. (14) to obtain

$$\begin{aligned} U_{Y(X)}^2 = & \sum_{i=1}^J \left(\frac{\partial Y}{\partial Y_i} \right)^2 B_{Y_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial Y}{\partial Y_i} \right) \left(\frac{\partial Y}{\partial Y_k} \right) B_{Y_i Y_k} + \sum_{i=1}^J \left(\frac{\partial Y}{\partial Y_i} \right)^2 P_{Y_i}^2 \\ & + \sum_{i=1}^J \left(\frac{\partial Y}{\partial X_i} \right)^2 B_{X_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial Y}{\partial X_i} \right) \left(\frac{\partial Y}{\partial X_k} \right) B_{X_i X_k} + \sum_{i=1}^J \left(\frac{\partial Y}{\partial X_i} \right)^2 P_{X_i}^2 \\ & + 2 \sum_{i=1}^J \sum_{k=1}^J \left(\frac{\partial Y}{\partial X_i} \right) \left(\frac{\partial Y}{\partial Y_k} \right) B_{X_i Y_k} \end{aligned} \quad (15)$$

where the partial derivatives are

$$\frac{\partial Y}{\partial X_i} = \frac{\partial m}{\partial X_i} X + \frac{\partial c}{\partial X_i} \quad (16)$$

and

$$\frac{\partial Y}{\partial Y_i} = \frac{\partial m}{\partial Y_i} X + \frac{\partial c}{\partial Y_i} \quad (17)$$

and where the partial derivatives can also be determined numerically, as discussed later.

Calibration Problem

If the value of X being used in the regression model is measured with the same apparatus as the X_i data, and thus shares the same systematic error sources, the uncertainty expression must be expanded to include these uncertainties and correlations. This problem is often encountered in calibration processes and the uncertainty expression is

$$\begin{aligned}
U_Y^2 = & \sum_{i=1}^J \left(\frac{\partial Y}{\partial Y_i} \right)^2 B_{Y_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial Y}{\partial Y_i} \right) \left(\frac{\partial Y}{\partial Y_k} \right) B_{Y_i Y_k} + \sum_{i=1}^J \left(\frac{\partial Y}{\partial Y_i} \right)^2 P_{Y_i}^2 \\
& + \sum_{i=1}^J \left(\frac{\partial Y}{\partial X_i} \right)^2 B_{X_i}^2 + 2 \sum_{i=1}^{J-1} \sum_{k=i+1}^J \left(\frac{\partial Y}{\partial X_i} \right) \left(\frac{\partial Y}{\partial X_k} \right) B_{X_i X_k} + \sum_{i=1}^J \left(\frac{\partial Y}{\partial X_i} \right)^2 P_{X_i}^2 + \\
& + \left(\frac{\partial Y}{\partial X} \right)^2 B_X^2 + 2 \sum_{i=1}^J \left(\frac{\partial Y}{\partial X} \right) \left(\frac{\partial Y}{\partial X_i} \right) B_{XX_i} + \left(\frac{\partial Y}{\partial X} \right)^2 P_X^2 \\
& + 2 \sum_{i=1}^J \sum_{k=1}^J \left(\frac{\partial Y}{\partial X_i} \right) \left(\frac{\partial Y}{\partial Y_k} \right) B_{X_i Y_k} + 2 \sum_{i=1}^J \left(\frac{\partial Y}{\partial X} \right) \left(\frac{\partial Y}{\partial Y_i} \right) B_{XY_i}
\end{aligned} \tag{18}$$

If no errors sources are common between the X and Y variables, the X-Y correlation terms (that comprise the last line of the above expression) are omitted.

Numerical Partial Derivative Determination

The partial derivatives can be approximated directly using a finite difference numerical technique, often referred to as a "jitter-routine."⁴ It was determined during this effort that round-off errors could be significant, so double precision arithmetic was used. Accuracy was also enhanced by using a second order accurate central-differencing scheme⁹, such as:

$$\frac{\partial r}{\partial X_1} = \frac{f(X_1 + h, X_2, \dots, X_n) - f(X_1 - h, X_2, \dots, X_n)}{2h} \tag{19}$$

Traditional Statistical Regression Uncertainties: Straight Line

To demonstrate the value of the uncertainty propagation methodology presented above, a comparison to the confidence intervals provided by traditional statistical techniques is needed. The form of the statistical intervals used in this study was taken from Natrella¹⁰, more discussion of these intervals and additional forms of the intervals can also be found in references [2] and [3]. The variance of the slope is given as

$$s_m^2 = \frac{s_y^2}{S_{xx}} \tag{20}$$

with the 95% confidence interval represented by

$$m \pm \lambda s_m \tag{21}$$

where the 95% coverage factor, λ , is determined using the two-tailed Student's t-distribution, with (n-2) degrees of freedom, as

$$\lambda = t_{.95}(n-2) \tag{22}$$

Similarly, the variance for the y-intercept is given by

$$s_c^2 = s_Y^2 \left\{ \frac{1}{n} + \frac{\bar{X}^2}{S_{xx}} \right\} \quad (23)$$

and the 95% confidence interval is determined as

$$c \pm \lambda s_c \quad (24)$$

The confidence interval associated with any point determined from the regression model is (what can be referred to as the predicted value) is obtained with the interval W_i as

$$W_i = \lambda s_Y \left\{ \frac{1}{n} + \frac{(X_i - \bar{X})^2}{S_{xx}} \right\}^{1/2} \quad (25)$$

and the 95% confidence region is expressed as

$$Y_i \pm W_i \quad (26)$$

Where the sum of squares statistics used in the determination of these intervals are defined as

$$S_{xx} = \sum_{i=1}^n X_i^2 - \frac{\left(\sum_{i=1}^n X_i \right)^2}{n} = \sum_{i=1}^n (X_i - \bar{X})^2 \quad (27)$$

and

$$S_{yy} = \sum_{i=1}^n Y_i^2 - \frac{\left(\sum_{i=1}^n Y_i \right)^2}{n} = \sum_{i=1}^n (Y_i - \bar{Y})^2 \quad (28)$$

and

$$S_{xy} = \sum_{i=1}^n X_i Y_i - \frac{\sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n} = \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) \quad (29)$$

and

$$s_Y^2 = \frac{1}{n-2} \left\{ S_{yy} - \frac{(S_{xy})^2}{S_{xx}} \right\} \quad (30)$$

X and Y as Functional Relations

In many, if not most instances, the test data will be expressed in some form of functional relationship. In these cases, the data will not be the variables used in the regression, instead the result from the functional relationships will be used. Examples of common functional relations are Reynolds number, flow coefficient, pressure coefficient, specific fuel consumption, etc. This can be expressed for a general example as

$$X_i = f(\text{VAR1}_i, \text{VAR2}_i) \quad (31)$$

$$\text{and } Y_i = f(\text{VAR3}_i, \text{VAR4}_i, \text{VAR5}_i) \quad (32)$$

so the expression for the uncertainty in the predicted value becomes

$$\begin{aligned} U_Y^2 = & \sum_{j=1}^5 \sum_{i=1}^n \left(\frac{\partial Y}{\partial \text{VARj}_i} \right)^2 P_{\text{VARj}_i}^2 + \sum_{j=1}^5 \sum_{i=1}^n \left(\frac{\partial Y}{\partial \text{VARj}_i} \right)^2 B_{\text{VARj}_i}^2 \\ & + 2 \sum_{j=1}^5 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial Y}{\partial \text{VARj}_i} \right) \left(\frac{\partial Y}{\partial \text{VARj}_k} \right) B_{\text{VARj}_i \text{VARj}_k} \\ & + 2 \sum_{j=1}^4 \sum_{l=j+1}^5 \sum_{i=1}^n \sum_{k=1}^n \left(\frac{\partial Y}{\partial \text{VARj}_i} \right) \left(\frac{\partial Y}{\partial \text{VARl}_k} \right) B_{\text{VARj}_i \text{VARl}_k} \end{aligned} \quad (33)$$

where the second term accounts for correlated systematic uncertainty sources within each variable and the third term accounts for systematic uncertainty sources common between variables. Similar expressions for the uncertainty in the slope and y-intercept are readily obtained. The complexity introduced by the transformation necessitates determining the partial derivatives numerically.

Another important form of transformation is when one or both of the variables are represented in logarithmic space. A similar technique is used whereby the functional relationship as expressed in Eq.s (31) and (32) become the logarithmic function desired, such as

$$X_i = \ln(\text{VAR1}_i) \quad \text{and} \quad Y_i = \ln(\text{VAR2}_i) \quad (34)$$

The partial derivatives then account for the transformation and the uncertainties in the measured variables are propagated in the units of the measurand without transformation.

Higher Order Regressions: Polynomial Curvefits

The general expression for a n^{th} order polynomial curve is

$$Y = a_0 + a_1 X + \dots + a_n X^n \quad (35)$$

where the regression coefficients, a_i , are determined with a least squares fit by minimizing the Chi-square function, χ^2 . The numerical method used in this work utilizes orthogonal polynomials and a singular value decomposition solution routine. A complete description of the matrix mathematics and numerical algorithm used in this work is provided in references [10] and [13]. The expression for the uncertainty in a value found using the polynomial curvefit is the same as in Eq. (15).

Monte Carlo Simulations

The methodology for assessing the uncertainty in linear regressions was developed using Monte Carlo-type simulations. The Monte Carlo simulations were conducted in the following manner.

"True" values for data from a relationship with specified coefficients were determined. The word *true* is emphasized to indicate that it represents the actual physical quantity of the parameter if it could be measured without any bias error or precision error, which is always an unobtainable value. The two-sigma (2 standard deviation or 95% confidence) bias limits and precision limits for each variable were then specified. The errors in each variable were assumed to come from these normally distributed error populations with the specified standard deviations. A random value for each bias error and precision error was found from a Gaussian random deviate generator subroutine using the specified standard deviations. The Gaussian deviates have a mean of zero and an equal probability of being positive or negative. The bias errors within each variable were assumed to be from the same source, and were assigned the same random deviate from the Gaussian distribution. In some cases these bias errors were a fixed amount and in some cases they were a percent of reading. Precision errors were obtained by sampling the precision error populations repeatedly to obtain independent random deviates for each variable. The individual error values, bias errors and precision errors were then summed and added to the *true* value to obtain a data point with errors from the specified error populations. These data points were then used in the linear least squares equations to obtain the value of the regression coefficients and the curvefit model. These coefficients and the curvefit model represent the "best fit" of the experimental data when the bias and precision errors are present.

A 95% confidence uncertainty interval for the result was calculated from the uncertainty propagation analysis equations for m , c , and Y . A $\pm U_m$ interval was placed around the slope coefficient value, m , and if the true value of the slope was found to be within the interval a counter was incremented. A similar procedure was used with the y-intercept and the predicted value from the curvefit expression. This procedure was repeated 10,000 times and the percent coverage, or number of times the true result was within the estimated interval, was determined. Using this procedure, the effectiveness of the uncertainty propagation equation could be investigated by checking whether or not the *true* value is within the 95% confidence uncertainty interval about the measured result 95% of the time, we can determine how well our uncertainty approximation equations work.

In the previous work¹ the precision uncertainties used in the propagation equations were determined using the "large sample approximation" and calculating a standard deviation at each setpoint. The effect of sample size, i.e. the number of setpoints and the number of data taken at each setpoint, was thus studied. The conclusion drawn from this analysis was that when the "correct" precision limit is used in the propagation equations the desired confidence interval is obtained. Thus, all the work in this report will assume that the correct precision uncertainties have been

estimated based upon prior information and experience so the simulations only evaluate the methodology.

A useful statistic from the simulation is the uncertainty ratio, the ratio of the average uncertainty intervals for the regression coefficients from the 10,000 iterations divided by the true 95% confidence intervals. The *true* 95% uncertainties are calculated as twice the sample standard deviations, S_m , S_c , and S_Y , from the 10,000 samples of the regression coefficients {not to be confused with the standard deviations of the curvefit, s_m , s_c , and W , respectively, as defined in Eq.s (20), (23), and (25)}. The sample standard deviations from the 10,000 sample population can be expected to be good representations of the actual standard deviations of the infinite population with the elemental uncertainty sources as defined. An uncertainty ratio near unity shows that the uncertainty methodology works for the particular case, with values greater than one meaning an overprediction and values less than one meaning an underprediction.

1st Order Regression Uncertainty Simulation and Results

The uncertainty associated with the value obtained from using a curvefit is determined using Eq. (15). The effectiveness of this expression was evaluated using the Monte Carlo simulation technique described above. The data defining the “true” relationship and the results from the simulation are shown in Tables 1, 2, and 3. Table 1 simulates the case where the systematic uncertainties are much greater than the precision uncertainties and where all of the systematic uncertainties are correlated, including an error source common to both variables. Table 2 presents a simulation where the systematic uncertainties are the same value as the precision uncertainties, and the systematic uncertainties are correlated within X and within Y, but not correlated between X and Y. The third simulation presented, Table 3, includes only precision uncertainties. In each table the average uncertainty interval, the percent coverage, and the uncertainty interval ratio is provided for the uncertainty propagation methodology and for the traditional statistical confidence interval equations.

Table 1 shows that the methodology provided the appropriate, i.e. ~95%, coverage for the predicted value and the regression coefficients, while the traditional statistical methods severely underestimate the uncertainty interval. Table 2 demonstrates that the methodology provides the appropriate uncertainty interval and the statistical methods do not provide the appropriate coverage. Table 3 presents the simulation when only precision, or random, uncertainties exist and it is seen that both methods provide the desired coverage, however the statistical methods significantly overpredict the uncertainty interval, by almost 50%. In all three cases the uncertainty ratio is near one, indicating the uncertainty propagation methodology produces appropriate uncertainty interval estimates.

Case 1: Dominant Systematic Uncertainties								
B _{X1} = 2.5 B _{X2} = 2.5% B _{X3} =B _{Y3} = 2.5								
B _{Y1} = 2.5 B _{Y2} = 2.5%								
P _X = 0.0 P _Y = 0.1								
			Uncertainty Propagation			Classical Statistics		
pt	X _i	Y _i	U _Y (avg)	Cov (%)	Ratio	W (avg)	Cov (%)	Ratio
1	80.0	90.0	4.58	95.4%	0.99	0.15	3.9	0.02
2	90.0	100.0	4.81	95.3%	0.99	0.11	2.5	0.02
3	100.0	110.0	5.06	95.4%	0.99	0.09	1.8	0.01
4	110.0	120.0	5.32	95.3%	0.99	0.11	2.3	0.02
5	120.0	130.0	5.59	95.4%	0.99	0.15	3.0	0.02
Coefficient Uncertainties			U _m	Cov (%)	Ratio	tS	Cov (%)	Ratio
true slope, m=1.0			0.036	95.6	1.01	0.005	21.1%	0.13
true y-intercept, c=10.0			3.56	95.3	0.99	0.47	20.2%	0.13

Table 1. 1st Order Regression Simulation and Results; Dominant Systematic Uncertainties.

Case 2: Comparable Uncertainties								
B _{X1} = 2.5 B _{X2} = 0.0% B _{X3} =B _{Y3} = 0.0								
B _{Y1} = 2.5 B _{Y2} = 0.0%								
P _X = 2.5 P _Y = 2.5								
			Uncertainty Propagation			Classical Statistics		
pt	X _i	Y _i	U _Y (avg)	Cov (%)	Ratio	W (avg)	Cov (%)	Ratio
1	80.0	90.0	4.47	95.3	0.99	3.42	85.3	0.89
2	90.0	100.0	4.02	95.4	0.99	2.41	77.6	0.70
3	100.0	110.0	3.87	95.4	0.99	1.98	71.4	0.60
4	110.0	120.0	4.03	95.4	0.99	2.42	77.4	0.70
5	120.0	130.0	4.47	95.6	0.99	3.43	84.9	0.89
Coefficient Uncertainties			U _m (avg)	Cov (%)	Ratio	tS (avg)	Cov (%)	Ratio
true slope, m=1.0			0.112	95.1	1.00	0.164	94.4	1.464
true y-intercept, c=10.0			11.84	95.0	1.00	16.5	94.0	1.394

Table 2. 1st Order Regression Simulation and Results; Comparable Systematic and Precision Uncertainties.

Case 3: Dominant Precision Uncertainties								
B _{X1} = 0.0 B _{X2} = 0.0% B _{X3} =B _{Y3} = 0.0								
B _{Y1} = 0.0 B _{Y2} = 0.0%								
P _X = 2.5 P _Y = 2.5								
			Uncertainty Propagation			Classical Statistics		
pt	X _i	Y _i	U _Y (avg)	Cov (%)	Ratio	W (avg)	Cov (%)	Ratio
1	80.0	90.0	2.74	95.6	1.00	3.15	95.0	1.46
2	90.0	100.0	1.96	95.3	1.00	2.25	94.8	1.46
3	100.0	110.0	1.59	95.3	1.00	1.84	94.6	1.46
4	110.0	120.0	1.98	95.4	1.00	2.28	94.9	1.46
5	120.0	130.0	2.76	95.4	1.00	3.17	94.8	1.46
Coefficient Uncertainties			U _m (avg)	Cov (%)	Ratio	tS (avg)	Cov (%)	Ratio
true slope, m=1.0			0.112	95.4	1.01	0.163	94.94	1.48
true y-intercept, c=10.0			11.31	95.5	1.01	16.5	94.79	1.47

Table 3. 1st Order Regression Simulation and Results; Dominant Precision Uncertainties.

Calibration Problem Results

The calibration problem was simulated, wherein the systematic uncertainty associated with the X variable in the curvefit expression was correlated with the systematic uncertainty in the X data used in the curvefit. Equation (18) was used with the X-Y correlation terms omitted. The same true linear expression was used as in the previous simulations, however the curvefit was evaluated

at calibration points between the curvefit data. The input data, uncertainty information, and simulation results are shown in Table 4.

Calibration Problem					
Case 1: Dominant Systematic Uncertainties					
B _X = 2.5		P _X = 2.5			
B _Y = 2.5		P _Y = 2.5			
			Uncertainty Propagation		
pt	X _i /X _i (cal)	Y _i	U _Y	Cov (%)	Ratio
1	80.0/ 85.0	90.0	4.24	95.2	1.00
2	90.0/ 95.0	100.0	3.91	95.2	0.99
3	100.0/ 105.0	110.0	3.87	95.0	1.00
4	110.0/ 115.0	120.0	4.12	95.0	0.99
5	120.0/ 125.0	130.0	4.68	95.1	0.99

Table 4. Calibration Problem Simulation and Results; Comparable Systematic and Precision Uncertainties.

Table 4 again demonstrates that the appropriate uncertainty interval is obtained with the proposed methodology. The regression coefficient results and the traditional statistics results are similar to Tables 1, 2, and 3. They have been omitted because they are not affected by the calibration problem.

Polynomial Regression Uncertainty Simulation Results

A 2nd order polynomial curve was used to generate “true” (X,Y) data pairs and the uncertainty propagation methodology was evaluated using the Monte Carlo simulation technique. The methodology provided the appropriate coverage for the curves simulated. Only a few different curves were modeled and simulated in this effort, and the type of curves simulated were not varied enough to produce conclusive evidence that the methodology will work in all curvefit applications. Thus, further work is required before the methodology can be fully recommended. In lieu of presenting the results of the polynomial simulation performed, the polynomial curvefit for the pressure coefficient versus flow coefficient will be discussed.

Application of Methodology to Compressor Stage Characteristics Maps

The methodology presented herein was applied to a generic compressor stage characteristic map used in the AEDC Dynamic Turbine Engine Compressor Code (DYNTECC) computer model¹². Two important compressor stage characteristic maps are the pressure coefficient (ψ^P) versus flow coefficient (ϕ) and the temperature coefficient (ψ^T) versus flow coefficient (ϕ). The compressor characteristics can be expressed in a number of ways, but the forms used in this study are

$$\Psi^P = (PR - 1) = \frac{P_{T2}}{P_{T1}} - 1 \quad (36)$$

$$\Psi^T = (TR - 1) = \frac{T_{T2}}{T_{T1}} - 1 \quad (37)$$

and

$$\phi = W_{cor} = \frac{W \left(\frac{T_{T1}}{T_{Tref}} \right)^{1/2}}{\left(\frac{P_{T1}}{P_{Tref}} \right)} \quad (38)$$

where the subscript T represents the total temperature or total pressure. As can be seen in Figure 1 the temperature coefficient versus flow coefficient map has a linear relationship over the normal operation (pre-stall) range and the pressure coefficient versus flow coefficient map has a polynomial relationship within that range. The normal operation range is the only region used within this study, although the methodology can be applied to the other operating regimes as well.

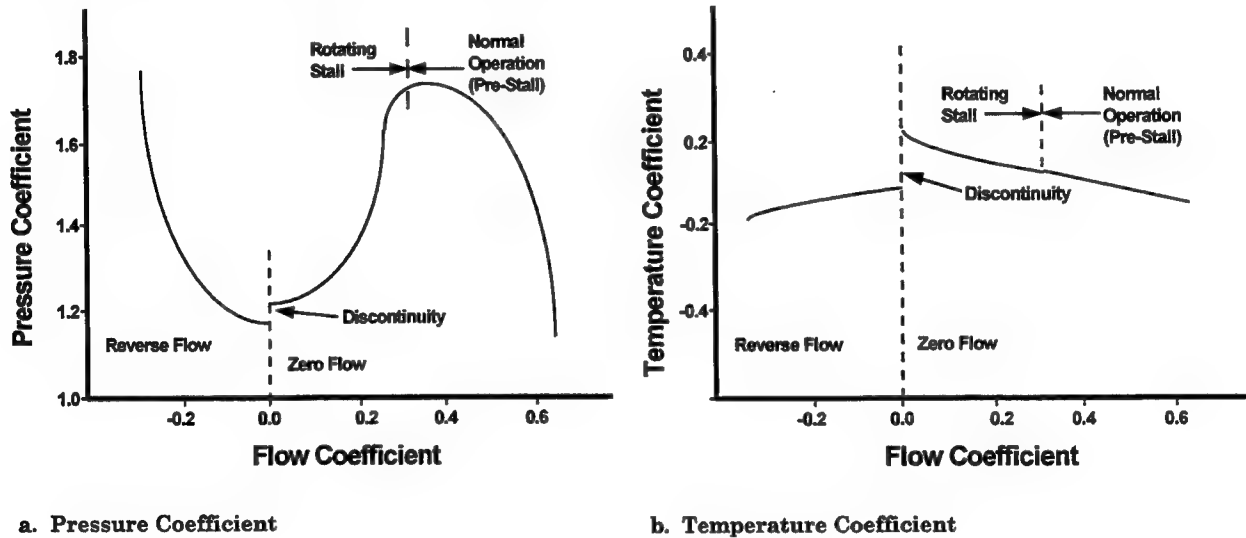


Figure 1. Typical stage characteristics.¹²

A straight line was used to model the normal operation region of the temperature coefficient map and a 3rd order polynomial was used to model the pressure coefficient map. Nominal values for each of the variables in equations (36), (37) and (38) were chosen that provided maps similar to those in Figure 1, and realistic uncertainty values for each of the measured parameters were chosen. It was assumed that all test data was taken with the same apparatus, so that the systematic uncertainties within each variable were correlated, and that the systematic uncertainties between the two temperature measurements and between the two pressure measurements were calibrated against the same standards, respectively, and using the same procedures so that they were also

correlated. A Monte Carlo simulation of the compressor map characteristics was performed using this input data to verify that the regression uncertainty methodology provided appropriate uncertainty intervals

The expressions for the uncertainty in the curvefit for the compressor stage characteristic maps are obtained by applying equation (33) to equations (36), (37), and (38). The expression for the temperature coefficient map is

$$\begin{aligned}
U_{\Psi^T}^2 = & \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial T_{T1}(i)} \right) P_{T1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial T_{T2}(i)} \right) P_{T2}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial P_{T1}(i)} \right) P_{P1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial W(i)} \right) P_W^2(i) \\
& + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial T_{T1}(i)} \right) B_{T1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial T_{T2}(i)} \right) B_{T2}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial P_{T1}(i)} \right) B_{P1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^T}{\partial W(i)} \right) B_W^2(i) \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^T}{\partial T_{T1}(i)} \right) \left(\frac{\partial \Psi^T}{\partial T_{T1}(k)} \right) B_{T1_i T1_k} + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^T}{\partial T_{T2}(i)} \right) \left(\frac{\partial \Psi^T}{\partial T_{T2}(k)} \right) B_{T2_i T2_k} \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^T}{\partial T_{T1}(i)} \right) \left(\frac{\partial \Psi^T}{\partial T_{T2}(k)} \right) B_{T1_i T2_k} + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^T}{\partial P_{T1}(i)} \right) \left(\frac{\partial \Psi^T}{\partial P_{T1}(k)} \right) B_{P1_i P1_k} \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^T}{\partial W(i)} \right) \left(\frac{\partial \Psi^T}{\partial W(k)} \right) B_{W_i W_k} + \left\{ \begin{array}{l} \text{correlation terms for } B_{T1P1}, B_{T2P1}, \\ B_{T1W}, B_{T2W}, B_{P1W} \text{ are neglected here} \end{array} \right\}
\end{aligned} \tag{39}$$

and similarly for the uncertainty in the pressure coefficient map characteristics

$$\begin{aligned}
U_{\Psi^P}^2 = & \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial T_{T1}(i)} \right) P_{T1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial P_{T2}(i)} \right) P_{T2}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial P_{T1}(i)} \right) P_{P1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial W(i)} \right) P_W^2(i) \\
& + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial T_{T1}(i)} \right) B_{T1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial P_{T2}(i)} \right) B_{T2}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial P_{T1}(i)} \right) B_{P1}^2(i) + \sum_{i=1}^n \left(\frac{\partial \Psi^P}{\partial W(i)} \right) B_W^2(i) \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^P}{\partial T_{T1}(i)} \right) \left(\frac{\partial \Psi^P}{\partial T_{T1}(k)} \right) B_{T1_i T1_k} + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^P}{\partial P_{T2}(i)} \right) \left(\frac{\partial \Psi^P}{\partial P_{T2}(k)} \right) B_{P2_i P2_k} \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^P}{\partial P_{T1}(i)} \right) \left(\frac{\partial \Psi^P}{\partial P_{T1}(k)} \right) B_{P1_i P1_k} + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^P}{\partial T_{P1}(i)} \right) \left(\frac{\partial \Psi^P}{\partial T_{P2}(k)} \right) B_{P1_i P2_k} \\
& + 2 \sum_{i=1}^{n-1} \sum_{k=i+1}^n \left(\frac{\partial \Psi^P}{\partial W(i)} \right) \left(\frac{\partial \Psi^P}{\partial W(k)} \right) B_{W_i W_k} + \left\{ \begin{array}{l} \text{correlation terms for } B_{P1T1}, B_{P2T1}, \\ B_{T1W}, B_{P2W}, B_{P1W} \text{ are neglected here} \end{array} \right\}
\end{aligned} \tag{40}$$

Neglecting the correlated systematic uncertainties as indicated in the above equations is a reasonable engineering assumption since the primary common error source between these variables would be from the calibration of the data acquisition system, which is typically small with respect to the other systematic uncertainties. These terms can readily be included, but they are neglected for simplicity. In these equations the flowrate is assumed to be a measured variable and not the result of a separate data reduction equation. Typically, the flowrate would be the result of several other measured variables and its data reduction equation would replace W in Eq. (38.), but was neglected for simplicity in this effort. Since the flowrate determination typically includes pressure and

temperature measurements, correlated systematic uncertainties probably exist and would have to be properly accounted for.

Realistic values for the precision uncertainties and the systematic uncertainties for each of the measured variables were chosen based upon typical instrumentation and calibration uncertainties. The true temperature coefficient map characteristic was modeled with the expression

$$\Psi^T = 0.2 - 0.5\phi \quad (41)$$

and the pressure coefficient map characteristic was modeled with the expression

$$\Psi^P = 0.63 + 6.3\phi - 9.9\phi^2 + 2.8\phi^3 \quad (42)$$

Ten sets of test data were generated using these expressions to represent the true compressor map characteristics and a Monte Carlo simulation was performed. The input test data and uncertainty estimates used are shown in Table 4, again note that the precision uncertainties for each variable used are the exact 2σ (two-standard deviation) values.

pt	T _{T1} (°R)	T _{T2} (°R)	P _{T1} (psf)	P _{T2} (psf)	W (lbm/sec)	ϕ	Ψ^T	Ψ^P
1	600.0	630.0	15870.00	42922.002	2.1147	0.300	0.050	1.705
2	600.0	615.0	15870.00	43520.301	2.4671	0.350	0.025	1.742
3	600.0	612.0	15870.00	43572.621	2.5376	0.360	0.020	1.746
4	600.0	606.0	15870.00	43612.055	2.6786	0.380	0.010	1.748
5	600.0	600.0	15870.00	43566.324	2.8196	0.400	0.000	1.745
6	600.0	597.0	15870.00	43512.188	2.8901	0.410	-0.005	1.742
7	600.0	592.5	15870.00	43392.647	2.9958	0.425	-0.0125	1.734
8	600.0	585.0	15870.00	43093.398	3.1720	0.450	-0.025	1.715
9	600.0	570.0	15870.00	42134.850	3.5245	0.500	-0.050	1.655
10	600.0	540.0	15870.00	38894.196	4.2294	0.600	-0.100	1.451

Table 5. Input Data for Compressor Stage Characteristic Maps

The systematic and precision uncertainties used in the simulation were:

$$B_{T_{T1}} = 2.0 \text{ } ^\circ\text{R} \quad B_{T_{T2}} = 2.0 \text{ } ^\circ\text{R} \quad B_{P_{T1}} = 150.0 \text{ psf} \quad B_{P_{T2}} = 150.0 \text{ psf} \quad B_W = 0.02 \text{ lbm / sec}$$

$$P_{T_{T1}} = 1.0 \text{ } ^\circ\text{R} \quad P_{T_{T2}} = 1.0 \text{ } ^\circ\text{R} \quad P_{P_{T1}} = 50.0 \text{ psf} \quad P_{P_{T2}} = 130.0 \text{ psf} \quad P_W = 0.085 \text{ lbm / sec}$$

pt	ϕ	Ψ^T	U_{Ψ^T} (avg)	Cov. (%)	Ratio	Ψ^P	U_{Ψ^P} (avg)	Cov. (%)	Ratio
1	0.300	0.050	0.0041	95.5	1.014	1.705	0.0215	95.8	0.998
2	0.350	0.025	0.0034	95.6	1.015	1.742	0.0168	95.2	0.985
3	0.360	0.020	0.0034	95.8	1.014	1.746	0.0170	95.2	0.986
4	0.380	0.010	0.0032	95.8	1.013	1.748	0.0176	95.3	0.988
5	0.400	0.000	0.0032	95.8	1.011	1.745	0.0184	95.4	0.989
6	0.410	-0.005	0.0032	95.7	1.009	1.742	0.0190	95.2	0.989
7	0.425	-0.0125	0.0032	95.6	1.007	1.734	0.0200	95.0	0.988
8	0.450	-0.025	0.0034	95.6	1.004	1.715	0.0225	94.6	0.985
9	0.500	-0.050	0.0040	95.4	0.999	1.655	0.0284	95.0	0.980
10	0.600	-0.100	0.0060	94.5	0.997	1.451	0.0456	94.8	0.972

Table 6. Compressor Stage Characteristic Map Simulation Results

The information in Table 6 shows that appropriate uncertainty intervals can be obtained for compressor map characteristics when correct estimates for the precision and systematic

uncertainties are used in the curvefit uncertainty methodology. Thus, the uncertainty associated with the temperature coefficient curvefit model or the pressure coefficient model is provided.

Figures 2a and 2b show the stage characteristics modeled and simulated and their associated uncertainty intervals. These figures demonstrate a single iteration from the simulation with the input variables having dominant systematic uncertainties. Dominant systematic uncertainties were used in this case to more clearly demonstrate the significance of properly accounting for systematic uncertainties. Figure 2b demonstrates that when systematic uncertainties are dominant, statistical confidence intervals only represent the “goodness of the fit” and do not contain the “true” value. While the uncertainty propagation interval properly accounts for the systematic uncertainties and contains the “true” value at approximately 95% confidence..

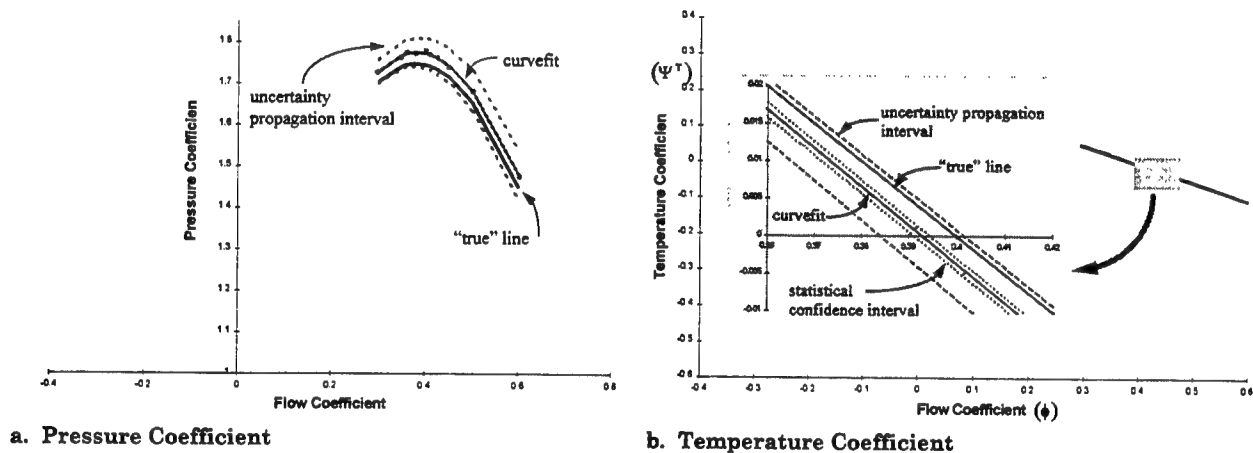


Figure 2. Compressor Stage Characteristic with Uncertainty Intervals
(Dominant Systematic Uncertainties)

In the development of the pressure coefficient simulation it was noted that the number of data sets and the spacing of the data was critical to obtaining a proper curvefit and appropriate uncertainty intervals. For example, if additional data sets were not clustered near the area of curvature in the pressure coefficient characteristic map, the curvefit obtained would not represent the defined curve. Also, in simulations where the errors were a sizable percentage, coverages greater than 95% were often obtained. Table 7 shows the results of a simulation where the uncertainties, both precision and systematic, were a large percent of the measurement, 12.5%. The methodology provided an overestimate of the confidence interval, around 99% coverage, with overestimates between 11% and 55%. These results and noting that a limited variety of possible curvefit scenarios were modeled indicate that further study to (1) verify the initial findings, (2) evaluate their impact, and (3) develop solutions be accomplished prior to recommending this methodology be applied to n^{th} order polynomial curvefits.

ϕ	ψ^P	$U_{\psi^P} \text{ (avg)}$	Cov. (%)	Ratio
0.300	1.705	1.225	99.1	1.11
0.350	1.742	0.532	98.8	1.18
0.360	1.746	0.479	98.8	1.23
0.380	1.748	0.420	98.8	1.36
0.400	1.745	0.409	99.2	1.47
0.410	1.742	0.418	99.4	1.51
0.425	1.734	0.445	99.4	1.55
0.450	1.715	0.532	99.2	1.55
0.500	1.655	0.939	98.2	1.34
0.600	1.451	4.199	97.1	1.16

Table 7. Compressor Stage Characteristic Map Simulation Results with 12.5% of Reading Systematic Uncertainties, Negligible Precision Uncertainties.

Future Research

A number of areas of continued research are identified as a result of this effort. In all of the Monte Carlo simulations performed, the precision limits used in the propagation equations were the exact (2σ) values. Work similar to that presented in reference [1] needs to be performed and extended to study the effect of sample size, the large sample approximation, data location, the use of prior information, and the use of other precision limit estimates. The polynomial curvefit results indicated that the methodology provides the appropriate uncertainty intervals in a well defined and well behaved problem. However, if the data and the curvefit using that data do not represent the "true" phenomena being investigated incorrect uncertainty intervals will be obtained. This could happen due to over-fitting or under-fitting the data. In general, the limitations of using this method to assess the uncertainties in polynomial curvefits must be investigated, and any underlying assumptions identified.

Since the primary motivation of this work is the assessment of uncertainties in propulsion system models, and since multivariate regression models are often used in engine system models a means of assessing the associated uncertainties is needed. Thus extension of this methodology to multivariate regressions is necessary.

Another important area for future research is the engineering application of this methodology. For example, how should it be integrated into data acquisition and test data reduction systems. How should the uncertainty information be documented so that others reading a report have the necessary information. And finally, an algorithm or flowchart describing how to implement this methodology would be very useful.

Conclusion

The work presented in this report provides a new methodology for assessing the uncertainty associated with curvefits of experimental information. This methodology is based upon the propagation of experimental uncertainties using accepted techniques. In the analysis it was shown

that the methodology provided the appropriate uncertainty intervals when correlated systematic uncertainties are present, and that traditional statistical confidence intervals do not provide the appropriate intervals in these situations. The methodology was shown to extend to cases where the curvefit variables are functional relations of experimental data. The methodology was applied to turbine engine compressor stage characteristics and provided the appropriate uncertainty intervals when fixed-value uncertainties were used. While the methodology was demonstrated for a 3rd order polynomial for the pressure coefficient map, a few areas were identified as requiring additional research to verify the robustness of the methodology.

References

1. Brown, Kendall K., Coleman, Hugh W., and Steele, W. Glenn, *Estimating Uncertainty Intervals for Linear Regression*, AIAA Paper 95-0796, 33rd Aerospace Sciences Meeting, Jan 1995.
2. Seber, G.A.F., *Linear Regression Analysis*, John Wiley & Sons, New York, 1977.
3. Montgomery, Douglas C. and Peck, Elizabeth A., *Introduction to Linear Regression Analysis*, 2nd Ed., John Wiley & Sons, New York, 1992.
4. Coleman, Hugh W. and Steele, Glenn W., *Experimentation and Uncertainty Analysis for Engineers*, John Wiley & Sons, New York, 1989.
5. *Guide to the Expression of Uncertainty in Measurement*, International Organization for standardization, Beneve, Switzerland, 1993.
6. "Quality Assessment for Wind Tunnel Testing," AGARD-AR-304, 1994.
7. American National Standards Institute/American Society of Mechanical Engineers, *Measurement Uncertainty*, PTC 19.1-1985, American Society of Mechanical Engineers, New York, 1986.
8. Brown, Kendall K., Coleman, Hugh W., Steele, W.G., and Taylor, Robert P., "Evaluation of Correlated Bias Error Effects in Experimental Uncertainty Analysis," *AIAA Journal*, (in press).
9. Burden, Richard L., and Faires, J. Douglas, *Numerical Methods*, 5th Ed., PWS Publishing Company, Boston, 1993.
10. Natrella, M.G., *Experimental Statistics*, NBS Handbook 91, National Bureau of Standards, United States Department of Commerce, 1963.
11. Press, William H., Flannery, Brian P., Teukolsky, and Vetterling, William T., *Numerical Recipes; The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1986.
12. Hale, A.A., and Davis, M.W., "DYNamic Turbine Engine Compressor Code DYNTECC - Theory and Capabilities," AIAA Paper 92-3190, AIAA/SAE/ASME/ASEE 28th Joint Propulsion Conference and Exhibit, Nashville, TN, July 1992.
13. Bevington, Philip R., and Robinson, D. Keith, *Data Reduction and Error Analysis for the Physical Sciences*, 2nd Ed, McGraw Hill, New York, 1992.

MODEL-BASED SYSTEM DEVELOPMENT
FOR
AEROSPACE TESTING INSTRUMENTATION

Dr. Theodore A. Bapty
Research Assistant Professor

James R. Davis
Graduate Student
Department of Electrical and Computer Engineering

Vanderbilt University
400 24th Avenue South
Nashville, TN 37235

Summer Faculty/Graduate Student Research Program
held at
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

September 1995

MODEL-BASED SYSTEM DEVELOPMENT
FOR
AEROSPACE TESTING INSTRUMENTATION

Theodore A. Bapty
Research Assistant Professor

James R. Davis
Graduate Student
Department of Electrical and Computer Engineering

Abstract

The complexity of modern aerospace systems requires significant amounts of testing to ensure correct operation and cost effective development. Ground testing enables extremely accurate control over the external conditions along with the ability to acquire a large number of measurements of internal engine structures.

Analysis and verification of turbine engine testing data is a computationally intensive, dynamic process. The combination of high bandwidths, large numbers of channels, and constantly changing processing requirements place many demands on an instrumentation and data analysis system. The complexity of building and managing the data systems necessary for the analysis/verification requirement is a difficult task, considering the real-time, parallel, and reactive characteristics of these systems.

Several systems have been developed at AEDC using **Model-Based** system synthesis techniques. This report describes three systems in various stages of development. The Computer Assisted Dynamic Data Acquisition and Monitoring System(CADDMAS), developed in conjunction with Arnold Engineering Development Center, has been in operation for two years in support of simulated altitude turbine engine stress testing. The system uses parallel processing to sustain over 800 million floating point operations per second(MFLOPS) producing on-line data analysis plots, delivering true supercomputer performance. Various improvements to the system have been implemented during the program.

The Data Validation (DATVAL) system, currently entering operation, monitors turbine engine performance data to verify data integrity. Errors are detected in the processed results and the sources of these errors are traced back to the input data, thus diagnosing the original problem. Methods for automatically generating the diagnosis information have been developed during the summer program.

A monitoring and diagnostics system for the large electric motors that power the testing facilities is currently in the initial design phase. The preliminary specifications for the system are currently under development.

The diverse systems described are developed using a common model-based program synthesis technique. These techniques benefit from software reuse and rapid application development. These factors greatly reduce the cost of the development efforts and decrease the design cycle time.

MODEL-BASED SYSTEM DEVELOPMENT FOR AEROSPACE TESTING INSTRUMENTATION

Theodore A. Bapty
James R. Davis

Introduction

During the development and operational lifespan of a turbine engine design numerous tests are performed to ensure design integrity and performance over time. Aerospace ground testing forms a basis for this evaluation process. The engine is installed in a controlled environment test cell that simulates a wide variety of conditions. The conditions are adjusted to approximate the entire operating envelope of the engine, determining the operation of the engine in the field.

Turbine Engine testing generates massive volumes of data at very high rates. A single engine may have several hundred sensors, with bandwidths up to tens of kilohertz. Test periods last for several hours. The entire bulk of data generated is on the order of terabytes for a single test.

Complete analysis of all acquired data has been technically impossible in the past, due to the speed and capacity of standard computers. Historically, only a small fraction of the data has been fully processed by periodic sampling of the data stream. This analysis was also performed off-line, with delays of days to weeks before results were available. This latency prohibits the use of results to guide the testing process.

On-line processing of these signals is critical to improving the testing process. The results from the on-line computations should be available to enable mid-test decisions and interactive test planning. Furthermore, on-line examination of information can assist in catastrophe avoidance, by enabling the detection of pathological engine conditions.

Performing these computations and displaying the results in an interactive format requires an extremely high processing rate. Typical algorithms include frequency transformation, peak detection, validation and correlation. For the analysis of 48 channels of stress data, approximately

800 MFLOPS of sustained computation is required. To process the hundreds of channels available in a typical test, an aggregate sustained computation rate in the GFLOPS range is necessary to meet requirements. The situation is further exacerbated by the requirement to acquire the volumes of data into the computer and to plot the results quickly. The aggregate data input is in the tens of megawords per second. Compounding these requirements, is the real-time requirement that the system must keep up with the data stream and display results interactively.

The majority of algorithms required for turbine engine testing are applied independently on each of the channels. The high computational requirements and natural parallelism make parallel processing a cost-effective hardware approach. The data streams are physically separate channels, so a parallel I/O system is also applicable.

To achieve the multi-GFLOPS goal, a unique computer architecture is required. The Computer Assisted Dynamic Data Monitoring and Analysis System (CADDMAS) has been developed to address these goals. The CADDMAS is a real-time, massively parallel processing system designed to process turbine engine stress data on-line. Results are available immediately, with system outputs defined interactively.

The massive volumes of data are recorded for post processing to assess engine performance and plan future system modifications. The integrity of the data from a test is of paramount importance to the success of the engine development. Hardware problems and configuration problems can result in data errors, corrupting the recorded data, eliminating the benefit of testing. Correcting these problems after the test is expensive, sometimes resulting in repetition of testing programs, at great fiscal and scheduling cost.

Consequently, the on-line verification of engine data is of utmost importance. The Data Validation (DATVAL) system, currently entering operation, monitors turbine engine performance data to verify data integrity. Errors are detected in the processed results and the sources of these errors are traced back to the input data, thus diagnosing the original problem. Representing the information needed for an accurate diagnosis is an extremely difficult task. Methods for automatically generating the diagnosis information have been developed during the summer program.

The facilities required to duplicate the necessary conditions to simulate high altitude flight are immense. Large compressors are required to pump large volumes air into and out of the turbine engine test cell. These compressors are driven by large, 30,000-80,000 horsepower electric motors. The motors have a complicated control system for startup and operation. These motors represent a significant capital expense and are critical to the mission of AEDC. Malfunctions in these motors

result in loss of testing revenue. Damage to the motors results in a large capital outlay and a long system downtime.

When the motors fail to start, a difficult diagnosis procedure ensues. This procedure relies heavily on the experience of small group of irreplaceable engineers and often takes several hours.

Predictive diagnosis could save significant resources, if an impending component failure can be anticipated. Preventive maintenance can concentrate on upcoming problems, reducing the occurrence of unexpected system failures. Preventive maintenance can also reduce the frequency of failure of expensive systems.

A diagnosis and predictive analysis system is currently being defined for use at AEDC. The approaches and capabilities are being considered.

While the hardware needed to meet these testing requirements is available for these systems, and the technology allows easy configuration of arbitrary topologies, the software technology is not as simple. The software is complicated by several factors:

1. **Complex Requirements:** The systems requirements consist of a complex set of processing specifications. The sheer number of specifications and their interrelations results in a significant management task.
2. **Changing Requirements:** The systems requirements change often. The changes must be integrated into the system rapidly.
3. **Distributed Control:** Instead of having one thread of control executing in the system, as many threads as there are processors are used in the system. The problem lies in getting all of these threads to cooperate and accomplish a single task.
4. **Communication:** As the computational data dependencies are divided across processors, there must be a method for transmitting intermediate results from one processor to another.
5. **Real-Time:** The data is coming into the system at a constant rate. The system must respond to all of the data at this rate, or data will be lost.
6. **Dynamically Changing Requirements:** The processing requested from the system is constantly changing. Each test point has a different mix of required output processing.
7. **Modularity:** Systems of various sizes and processing capacity are required. The components (hardware and software) must be designed to allow modules to be assembled quickly and easily

to form arbitrary systems.

8. **Cost:** Systems must be constructed for little more than the hardware cost, due to the cost of the high performance hardware needed to meet system performance goals. Historically, software accounts for 3/4 of the cost of a system like the CADDMAS.

These problems must be solved in a way that will not make the system implementation cost-prohibitive to implement. The next section describes the Model-Based synthesis approach. This discussion is followed by a description of the CADDMAS, the DATVAL, and the prototype engine diagnosis.

Model-Based Synthesis: Modeling the Problem

Multiple-aspect model-based systems offer significant advantages in the management of large-scale, complex systems:

1. Modeling allows abstractions of details into understandable concepts. The model can show only the important information for the current mode of operation. Low-level details can be hidden from the user, and system information can be extracted from these models using information about the modeling domain.
2. Multiple aspect model representation allows the interaction with the model from different perspectives at different times. The user only has to deal with the current mode.
3. Hierarchy in the models enables the hiding of structural details. Only relevant information is visible at any one level in the hierarchy. Complex structures can be represented by a single icon. The user can traverse this hierarchy to examine the system in greater detail.

Domain-specific modeling extends these benefits. The capability to adapt the model representation schema to the application domain further simplifies the problem by presenting familiar concepts to the user. Large systems can still be modeled, but the complexity is hidden in abstractions, alternate aspects, and/or hierarchies.

The selection of the modeling concepts and representation schemas is a key factor in the success of a model-based system. The development of efficient concepts requires a detailed knowledge of

the application domain, the model representation methods, and the interpretation process that maps between the models and the executable system.

The modeling specification and design process begins with the recognition of responsibilities and capabilities of the future system users. Typically, several classes of users will interact with the system. For example, consider the case of the parallel instrumentation domain. The following classes of users are apparent:

- **Instrumentation System Engineers(ISE)** are responsible for development of inherent system capabilities. These include low-level software development, algorithm implementation, and configuration of all system inputs and outputs. The ISE has in-depth knowledge of signal processing, software development, and hardware configuration. These constraints can be relaxed. For instance, when the software is taken from a reusable library, software development expertise is not required.
- The **Test Engineer** is responsible for configuring a working system and handling rapidly changing specifications. The specifications can be changed up to the point of system use, and possibly even during system operation. The test engineer has a deep understanding of the application, and can determine the proper set of requirements to place on the instrumentation system. The test engineer, however, has little knowledge of digital signal processing, software engineering, parallel processing, and embedded systems. This user deals with the concepts of number of system inputs, number and capabilities of system outputs, and requirements for system performance (timing, precision, etc).
- The **Test Operator** actually uses the system, gathering outputs and interpreting the results. This user sees the instrumentation system as a pre-configured black box that generates useful information. The Operator deals with the concepts of a Graphical User Interface for easy access to data, previously configured analysis scenarios, and, most importantly, analysis of the Instrumentation system's output.

The structure of a successful modeling paradigm must recognize the environment: inputs, outputs, system architecture, and the user classes. In this pursuit, the architecture for model-based synthesis of parallel instrumentation systems has been developed. The structure of this paradigm is seen in figure 1.

The Instrumentation Systems Engineers interact with the system to define the library of possible system outputs, using the System Capability Modeling Environment. For each type of analysis,

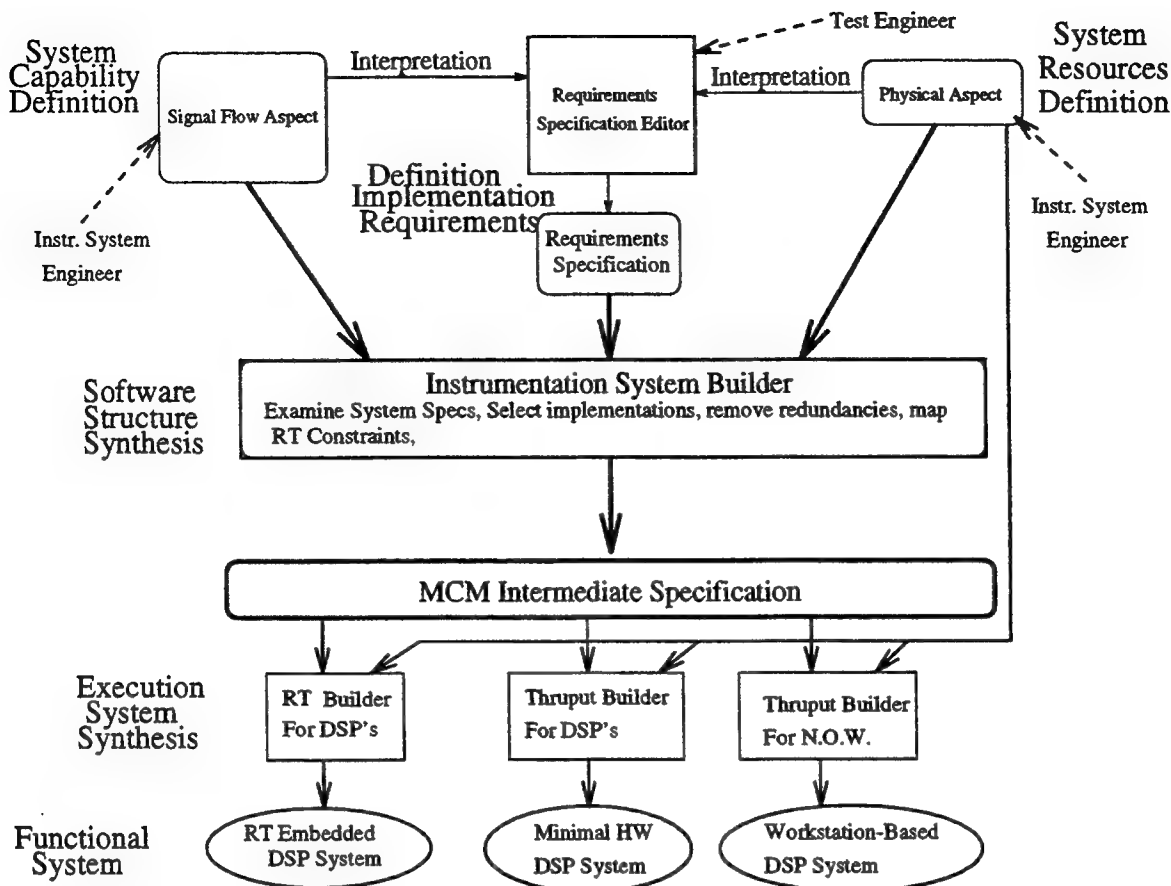


Figure 1: Real-Time Instrumentation System Synthesis Environment

models are defined representing the output type, the various algorithmic implementations of the analysis (in terms of a set of signal flow graphs), and a method for selecting and configuring the various implementations. These signal flow graphs are linked to the source code implementations of the algorithms.

The System Resource Modeling Environment allows modeling of the available input signals, hardware resources, and user interfaces available for system interaction.

These models can be interpreted to create a third modeling environment. This implementation requirements modeling environment allows the test engineer to model a specific instance of an instrumentation system using the available algorithmic, signal, and hardware resources. This environment is a drastically simplified, menu-based interface that allows the test engineer to interact with the systems development process in a natural manner.

Model-Based Synthesis: Synthesizing Systems

Figure 2 represents the typical design flow for a model-based instrumentation system. The process will be described in general in this section, in order to give the overall structure of the process. Each of these phases will be described in detail in the following sections.

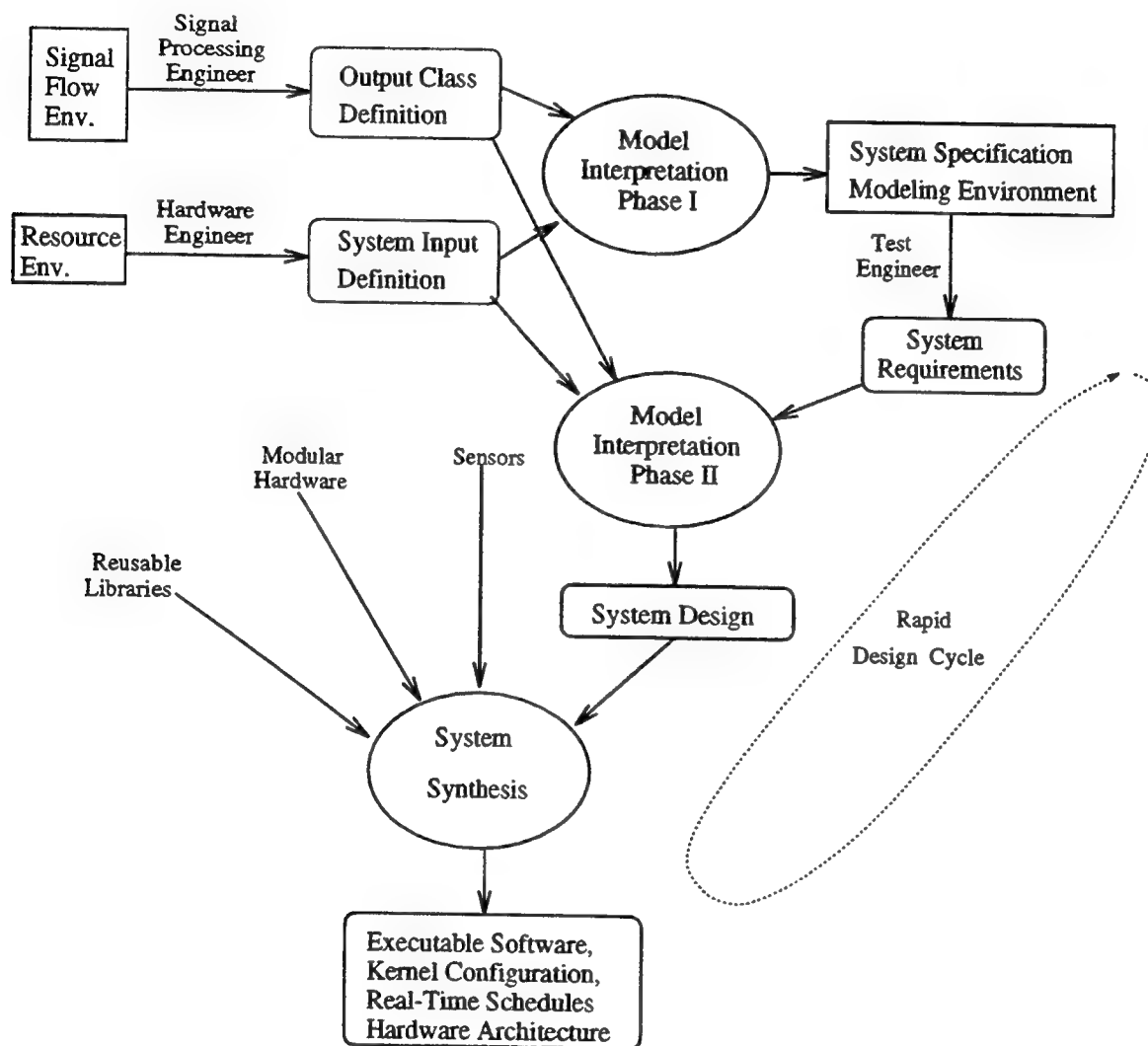


Figure 2: Real-Time Instrumentation System Design Cycle

In phase I, the Instrumentation Systems Engineer defines the capabilities and system inputs, using the signal flow modeling environment. The results of this modeling effort are the Output Class Definition and the System Interface Definition. The Output Class Definition process occurs

relatively infrequently. It is analogous to the generation of function libraries in standard software engineering practice. The relevant sets of system inputs, hardware resources, and system outputs are grouped together into a system structure model.

Phase I entails an interpretation of the component definitions referenced in the system structure model to create an environment for the Test Engineer, the System Requirements Specification Environment. The creation of this environment is analogous to compiling and documenting the function libraries.

The remainder of the design process occurs often: i.e. every time the system requirements change. During phase II, the System Requirements Specification Environment is used by the test engineer to create a requirements model database. This database is interpreted along with the output class database to generate a system software design, using a high-level system specification formalism.

In phase III, the system design specification is used, along with the hardware definition objects referenced in the system structure model, to synthesize a combined hardware/software architecture. The output is a hardware design, along with a scheduled, allocated, executable network of software modules, for the various processors. These outputs are used to build a parallel network and configure a modular kernel. The resultant structure then executes the program matching the requirements specified after Phase I.

The Phase II-Phase III process occurs rapidly, to meet changing user requirements. This cycle can also be used to support dynamically changing systems, with reinterpretation occurring online.

Application Example #1: The CADDMAS System

The CADDMAS consists of a loosely coupled set of processors, connected in an application-specific topology. This section will discuss the physical architecture of CADDMAS along with the functional distribution of tasks being executed on the processors.

Figure 3 show the overall CADDMAS architecture. The architecture is scalable to large numbers of channels and user interfaces. Each user interface has the same set of capabilities, and each channel is processed in a uniform manner.

The various processors in the system and their functions are:

- **User Interface:** The user interface hardware consists of a 386-or-better PC with a high

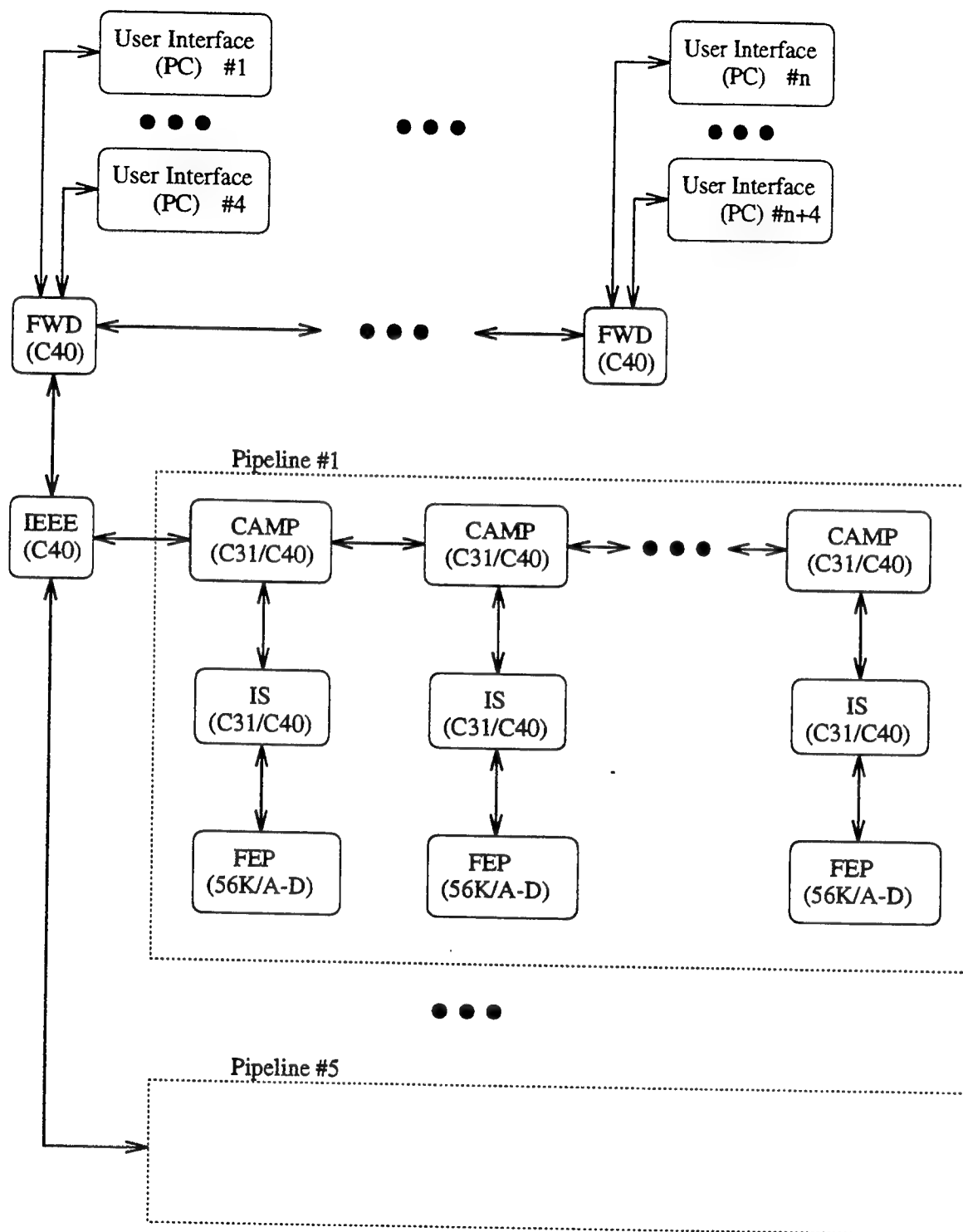


Figure 3: C40/C31 Architecture Diagram

resolution monitor, accelerated graphics card, keyboard, mouse, and optional printer. One of the user interfaces resets, bootstraps, and loads the network of processors with the proper executable software on system startup. All user interfaces execute a graphical user interface program with the following functions:

1. Accept user input to allow configuration of plot screens and selection of data sources.
2. Transmission of commands to the system of parallel processors to configure datapaths, request data, and control data processing modules, such as campbell and spectral envelope collection.
3. Acquire data from the system and maintain real-time updated copies.
4. Plot graphs as requested in the user interface configuration.
5. Print graphs to the local printer.
6. Store snapshots of data to the local disk.

The user interfaces also use the parallel system as a local user-interface network. Via this network, the user interfaces can update each other, with screen configurations, datapoint labels, and print/store commands. This facility allows a single user to control the operation of multiple user interfaces.

- **FWD:** The Forwarder serves as a data control and distribution network for the set of user interfaces. User interface commands propagate through these nodes to the network of processors, and are interpreted en-route. Tables of data requests are maintained for each user interface, allowing efficient routing of data packets. Data is acquired from the network via the IEEE processor and shipped to the proper set of user interfaces.

The data transmission function of these processors leaves the other CPUs lightly loaded. Thus, these processors are available for expansion of the CADDMAS algorithms.

- **IEEE:** The IEEE serves as a data routing processor, interfacing the network of User Interface/FWD processors to the network of data processing nodes. The data is merged from several processing pipelines. On a T800, 3 channels are available, while 5 are available on a C40 system. The commands from the user interfaces are distributed to each of the pipelines. The IEEE processor also converts data formats, since the C31/C40 processors use TI floating point format and the PC uses IEEE.

- **CAMP:** The Campbell processor performs the collection of spectral peak information into a database to be used for plotting campbell diagrams. This processor also gathers the data for the Engine Order and the RPM-Time tracking. In addition, the Campbell processor implements data forwarding up the pipeline towards the IEEE processor.
- **IS/IS_DB:** The IS processor (in the case of the C40) and the combination IS/IS_DB processors perform the following: **Engineering Unit (EU) Conversion, Complex Spectrum Computation, Spectral Magnitude/Envelope Computation, and Spectral Peak Detection.**

These computations are done directly on the C31 or C40 for TI-based systems. For transputer systems, the Zoran VSP performs the majority of the mathematical operations.

- **FEP:** The FEP samples the analog data at a frequency 2.56 times the system bandwidth and transmits out a link compatible with the processing nodes in the system.

Data Types

The information flowing throughout the system is characterized by a limited set of data types. These types are described below: **Raw Data:** The raw, integer data comes directly from the sampling system. **Time Data:** Time-domain data represents physical quantities converted into appropriate units. **Complex Spectral Data:** the complex spectral estimation resulting from a real-input Fast Fourier Transform(FFT). **Spectral Magnitude Data** is the absolute value of the complex spectral data. **Spectral Envelope Data** is the peak-hold envelope of the spectral magnitude data. **Spectral Peaks Data** is the peaks of the spectral magnitude data. **Campbell Data** is a two-dimensional database of spectral magnitude peaks, spectral phases, and correlated parameters collected over time within a range specified by the user. The data is collected, collated, and filtered over an entire test point. The driving parameter is commonly engine RPM, but time or any other parameter is valid. **Engine Order Data:** is a one-dimensional database corresponding to the peak spectral magnitude at a specific multiple of engine RPM (or any other measured parameter). The independent variable is the RPM/parameter. **RPM-Time Tracking Data** is similar to the Engine Order data, except using the maximal time-domain peak as a function of parameter.

CADDMAS Application Areas

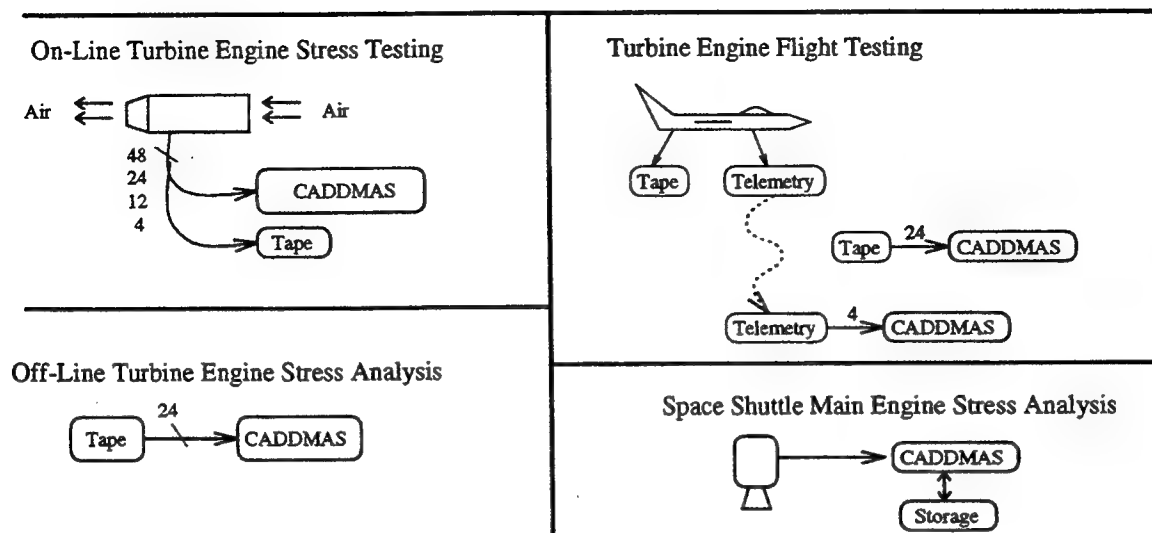


Figure 4: Applications of CADDMAS

CADDMAS Systems in Operation

The CADDMAS architecture has been employed in several systems (see figure 4). Both major engine test facilities have large permanent systems installed. The first facility uses a 64 processor, 2 GFLOP CADDMAS system based on the TMS320C40 and TMS320C31 processors. 5 interactive user interfaces are supported concurrently, processing 48 channels of data. The second system is based upon the Inmos T800, using approximately 36 processors. Several other systems are in operation: A flight test facility is using a small 12 channel CADDMAS system for online analysis of flight data. An extended CADDMAS is under development for NASA for use on the Space Shuttle Main Engine. This system incorporates advanced algorithms for vibration analysis.

Application Example #2: The DATVAL System

The steady-state data validation process at AEDC involves multi-step data acquisition and processing, that must be performed before the results of the validation process can be shown. Data is collected from sensors placed on a test article and is sent to a large (approximately 50,000 lines of source code) data reduction program. Output data from the data reduction program can then

DatVal Data Flow

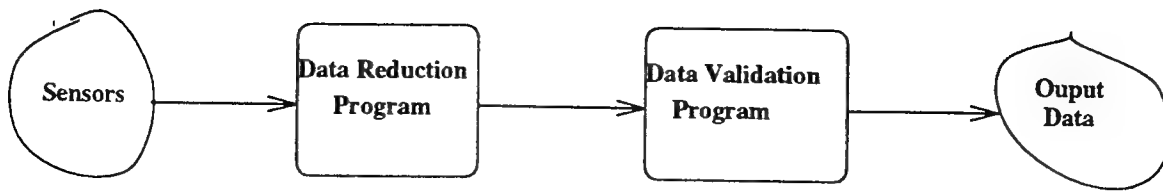


Figure 5: Data validation data flow

be fed to the actual data validation program (See figure 5). Currently, the data validation program lacks several features; no graphical user interface, no diagnostics, and limited visualization of the data are some of areas that need to be upgraded.

Data is sampled from the engine in blocks known as data points. Each data point contains several steady-state measurements from the engine. These data points arrive approximately every two minutes, but can arrive in much shorter intervals. Each data point must be processed to ensure the quality of the data before it is stored on magnetic tape, for delivery to the engine manufacturer.

A new data validation package (DATVAL), which is currently being brought on-line, uses a graphical interface and many added data viewing features to help the test engineer validate test data. Some of the new features include the ability to look at the data in its reduced form, the ability to view the data in the internal formats (with further reduction/computation performed) of the data validation package, tolerances for key data sets can be modified on-line by the test engineer, and rudimentary diagnostics can be performed. One of the key advantages of this new system is the ability to look at any and all data at any given time. With this new system, all of the functionality of the old system is maintained, and these several features only add to the usefulness of the system.

DatEdit, the DATVAL system editor, allows modification of the system on-line. This is the model editor (and run-time system synthesizer) used for the DATVAL system. It differs significantly from the model editor used for the CADDMAS system in several minute ways. Using this package, a test engineer can make changes to the system and, in a very short period of time, see the results of those changes. DatEdit uses the original data validation routines as part of the system model used to generate the run-time system. By easily allowing changes and automatically generating the DatVal system (run-time system) from the system models, DatEdit allows the test engineer great flexibility in the data validation process. If the engineer has a need to add more tests to the data validation system, he can add these new features with DatEdit during a test period.

Then, before the next data point arrives, these changes will be in effect.

Using model based techniques allows an engineer who is not trained in computer systems to build a custom, complex application in a very short amount of time. One of the goals of the DATVAL project is to allow a test engineer to customize the data validation process to enhance his ability to ensure the data is correct. DATVAL is a diversion from the other model based systems in one key area: the use of legacy code. By requiring the use of the original data validation routines, a new method for dealing with the model information contained in these routines had to be devised.

One of the problems with the DATVAL is the first-time system configuration. DATVAL uses the old data validation routines internally, adding a "wrapper" of functionality around the old routines. While building a system that uses these old routines is not very difficult, to take full advantage of the additional features of DATVAL additional information must be supplied. The set of internal variables and input variables that influence each check must be specified. Key data items such as tolerances must be specified separately, so as to allow the on-line modification of these items. Currently, this information must be entered manually.

A procedure has been developed that can extract most of this information automatically. This information is included in the model of the engine. Unfortunately, only the source code contains the model information. By extracting the data dependency information from the source code, the engine model data can be derived. A data dependency analysis will return a tree type structure that maps all of the inputs of the data validation program to their corresponding outputs. This information can be used for automatic system configuration.

A data dependency analysis tool (DDAT) has been developed. Currently, this tool can determine approximately 90% of the data dependency information contained in the data validation routines. Some small changes are needed to increase the effectiveness of this tool. After these changes have been made, testing can commence on the data reduction program. Work on the data reduction program has not commenced due to the large size of the program. Until the DDAT has been shown to effectively analyze the data validation routines, it will not be used on the data reduction program.

DDAT is a multipass analyzer. First, a package called Sage++ is used to parse the fortran source code into an object database used by DDAT. Several passes through the object database then occur. The first pass compiles all the common block variables into specific data structures. Next, the mapping of the input variables to the common block variables is extracted. Then, the mapping from internal variables to outputs is extracted. This allows a tree to be constructed that maps from the reduced data to the data error displays. When the data reduction program can be

analyzed, a complete mapping of outputs to sensors can be produced.

By using the DDAT, the DATVAL configuration will be much easier and less labor intensive, allowing the DATVAL package to be used much more efficiently. Currently, the DDAT is undergoing testing and modification. Modifications are being made to correct shortcomings in the algorithm that is used to extract the data dependencies. This algorithm is constantly changing to incorporate new features that improve upon the usefulness of DATVAL.

Application Example #3: The Motor Monitoring/Diagnosis System

The motor monitoring/diagnosis system uses the model-based architecture to represent the structure of the system under diagnosis. The modeling paradigm captures the physical and electrical structure of the motor and controller. From the electrical structure, a partial diagnosis approach can be derived, based on component functions and connection topology. Additional diagnostics information is to be modeled within the environment, to represent other physical interactions and accumulated experience from previous system diagnosis exercises.

From these models, a real-time diagnosis system is to be derived. (See figure 6). This diagnostics system interfaces directly to sensors on the motor, compressor, and controller. Timed readings of these sensors form the basis for the diagnostics process.

Additional predictive diagnosis processes are to be developed by watching trending information. For example, the trends of the current-voltage motor startup curves may reveal gradual changes in system performance. Analysis of current traces for insulator breakdown can also predict incipient insulator breakdown and subsequent coil damage. These are a small sampling of the analysis possible. The model-based approach captures the knowledge about the physical plant necessary to rapidly implement various diagnostics systems.

Acknowledgments

The authors acknowledge the kind support of the United States Air Force in making possible the work described in this paper as well as the CADDMAS Team in addition to the authors:

- James Mitchell: Arnold Engineering Development Center, Directorate of Technology.

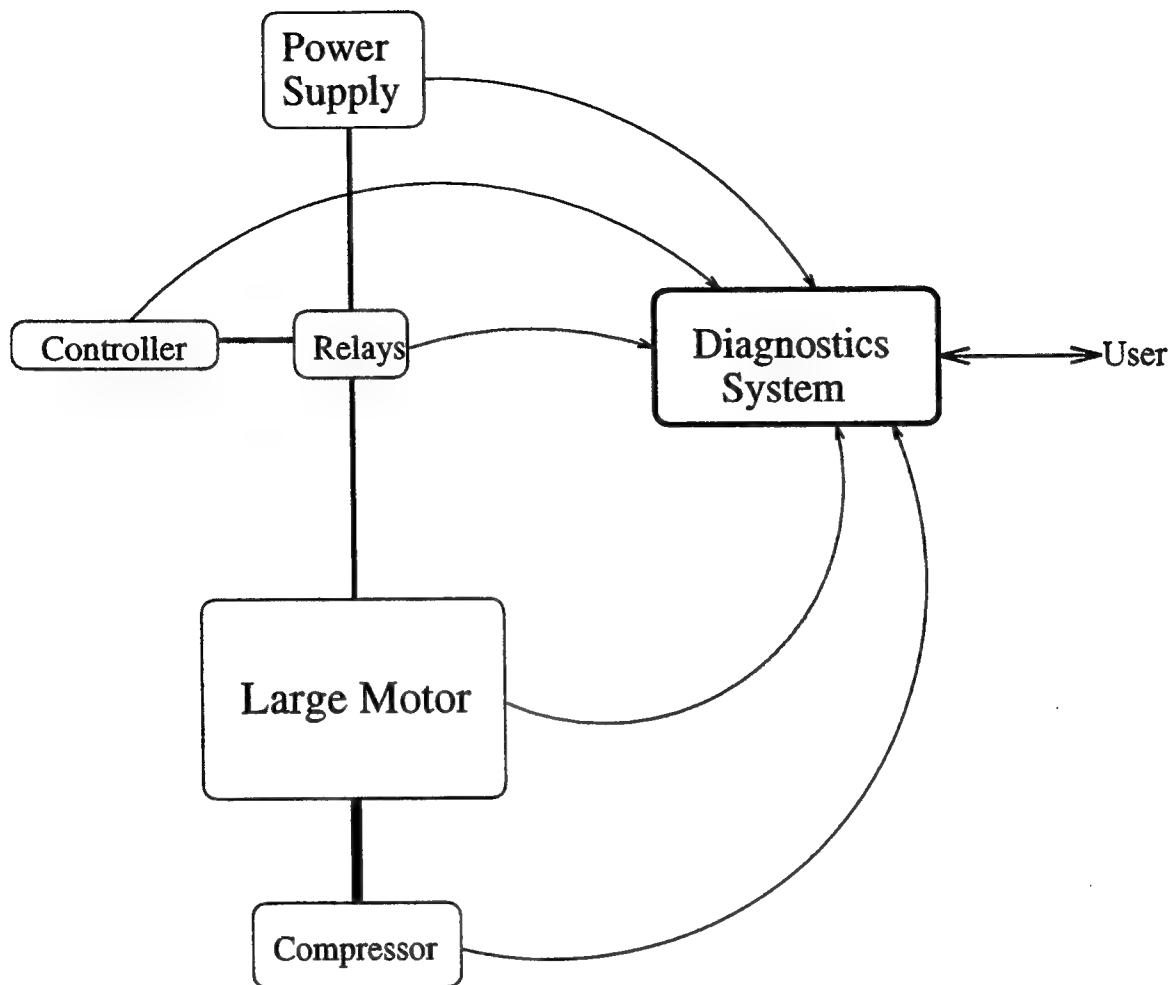


Figure 6: Simplified Motor System Diagnosis Architecture

- Tom Tibbals, Terry Hayes: Sverdrup Technology, AEDC Division
- Dr. Bruce Bomar: The University of Tennessee Space Institute

References

- [1] Bapty, T.: "Model-Based Synthesis of Real-Time Systems" Ph.D. Thesis Dept. of Electrical Engineering, Vanderbilt University, Nashville, TN., Dec. 1995.
- [2] Abbott, B, Bapty T, Biegl, C, Karsai, G., Sztipanovits, J. "Model-Based Approach for Software Synthesis", IEEE Software, pp. 42-53, May, 1993

- [3] Karsai, G., Sztipanovits, J. "A Visual Programming Environment for Domain-Specific Model-based Programming.", IEEE Computer, May 1995.
- [4] Ledecz, A., Bapty, T., Karsai, G., Sztipanovits, J., "Australian Computer Journal ", In Press.
Sztipanovits, J., Abbott, B., Bapty, T., Misra, A., "Model-Based Synthesis of Complex Embedded Systems", Proc. of the 1994 Complex Systems Engineering Synthesis and Assessment Technology Workshop, Washington D.C. July 19-20, 1994
- [5] Bapty, T.A., Abbott, B.A., Biegl, C. Ledecz, A. "Parallel Turbine Engine Instrumentation System", Proc. of the 9th AIAA Conference on Computing in Aerospace, San Diego, CA, October, 1993
- [6] Ledecz, A. Abbott, B. Bapty, T., Biegl, C., "Parallel DSP System Integration", Microprocessors and Microsystems, Vol 17, No. 8. 1993.
- [7] "Research on Intelligent Process Control Systems" Dept. of Electrical Engineering, Vanderbilt University, Technical Report #88-003, 1988.
- [8] F. Bodin, P. Beckman, D. Gannon, J. Gotwals, S. Narayana, S. Srinivas, B. Winnicka, *Sage++: An Object Oriented Toolkit and Class Library for Building Fortran and C++ Restructuring Tools*, Proc. Oonski '94, Oregon, 1994.

**DEVELOPMENT AND EVALUATION OF A NUMERICAL MODEL FOR
ANALYSIS OF AERODYNAMIC ICING SIMULATIONS**

**Brian C. DeAngelis
Graduate Assistant
Department of Aeronautical and Astronautical Engineering**

**University of Illinois at Urbana-Champaign
104 South Wright Street
Urbana, IL 61801-2997**

**Final Report for:
Graduate Student Research Program
Arnold Engineering Development Center
Arnold Air Force Base, TN**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

**Arnold Engineering Development Center
Arnold Air Force Base, TN**

August 1995

DEVELOPMENT AND EVALUATION OF A NUMERICAL MODEL FOR
ANALYSIS OF AERODYNAMIC ICING SIMULATIONS

Brian C. DeAngelis
Graduate Assistant
Department of Aeronautical and Astronautical Engineering
University of Illinois at Urbana-Champaign (UIUC)
Urbana, IL 61801-2997

Abstract

The objective of this research was to further develop methodology previously designed for the numerical modeling of wind tunnel icing simulations. The KTVA-II code had been modified for efficiency by accounting for assumptions noted for these types of flows. Using this code, a reduced vapor model was tested and implemented at significant computational cost savings. It was determined that vapor pressure in these types of flows has negligible computational effect on the evaporation of water droplets, leading to the subsequent removal of all vapor calculations. An acceleration model was also tested to improve calculation speed, at a mild loss in accuracy. Due to the level of accuracy required from these computations, however, the acceleration scheme could not be implemented. The methodology was validated using experimental data generated for a variety of particle types experiencing dispersion in a decaying turbulence channel flow. Validation of another data set generated for an icing-type spray and flow field is currently progressing. Finally, a unified icing simulation code named K-ICE was designed based on results from these studies.

DEVELOPMENT AND EVALUATION OF A NUMERICAL MODEL FOR ANALYSIS OF AERODYNAMIC ICING SIMULATIONS

Brian C. DeAngelis

Introduction

Arnold Engineering Development Center (AEDC) is currently conducting altitude engine testing in the Aeropropulsion Systems Test Facility (ASTF) wind tunnel and reproducing engine icing conditions by the introduction of a super-cooled micro-droplet cloud. The icing cloud is generated by a series of upstream spray bars, each supporting several water spray nozzles. Evaluation of engine performance in such an environment is crucial for demonstration of air-worthiness as defined by FAA regulations.

In practice, the mechanisms governing the cloud quality at the engine face of an engine test facility (ETF) are known only in general. It is desired to create a uniform liquid water content (LWC) at the engine face. LWC is defined as the mass of water per unit volume of air. AEDC has sought a means of reproducing the icing flow field in the ASTF wind tunnel to more efficiently test the best methods of achieving uniform LWC at the engine face. Also, more recent applications of ice ingestion into rotating turbomachinery (personal communication with Dennis Lankford) has further provided the impetus for such studies.

This research represents continuing investigations into development and evaluation of a numerical model for analysis of aerodynamic icing simulations. Loth (1993) proposed a methodology by which the air flow field of a test facility would be calculated with a $k-\epsilon$ code such as NPARC (Cooper and Sirbaugh, 1989), followed by a droplet trajectory calculation performed by a modified version of KIVA-II (Amsden, *et al.* 1989). KIVA-II is an internal combustion engine code developed at Los Alamos which uses an arbitrary Lagrangian-Eulerian (ALE)

scheme to solve the fully-coupled, equations of motion for a fuel spray and gas in a combustor or non-combustor field. It is also a k- ϵ code, which lends itself to compatibility with NPARC. Loth (1993) noted that, due to the nature of icing simulation flows, certain simplifying assumptions could be made to improve the efficiency of the calculations. Among these, low liquid mass loading allows the coupling of the gas and droplet phases to be reduced from a two-way to a one-way coupling. This means that the gas phase affects the droplets, but the droplets do not significantly affect the gas, enabling the separate calculations of the gas and then the droplets using the gas solution.

Work this summer was completed in four phases. These are:

1. parametric studies of evaporation parameters
2. development and testing of acceleration schemes
3. validation of current methodology
4. creation of a unified icing code

This paper will discuss the theory and results from each of these phases.

Parametric Studies of Evaporation Parameters

Several droplet trajectory calculations were performed by the modified KIVA-II code with varying flow parameters. These parametric studies were completed to determine the essential mechanisms governing the dynamics of the droplets and the optimal combination of computational parameters to achieve these solutions. Droplet computations were made using a portion of the air flow field for the ASTF wind tunnel calculated by NPARC. A slice of the entire quarter-section of the ASTF air flow calculation between spray bars 1 and 2 was extracted along existing grid lines. Figure 1 shows this extraction. The computational boundary created at the maximum x-plane was treated as a symmetry boundary, since streamlines were nearly always parallel to this plane. All side walls

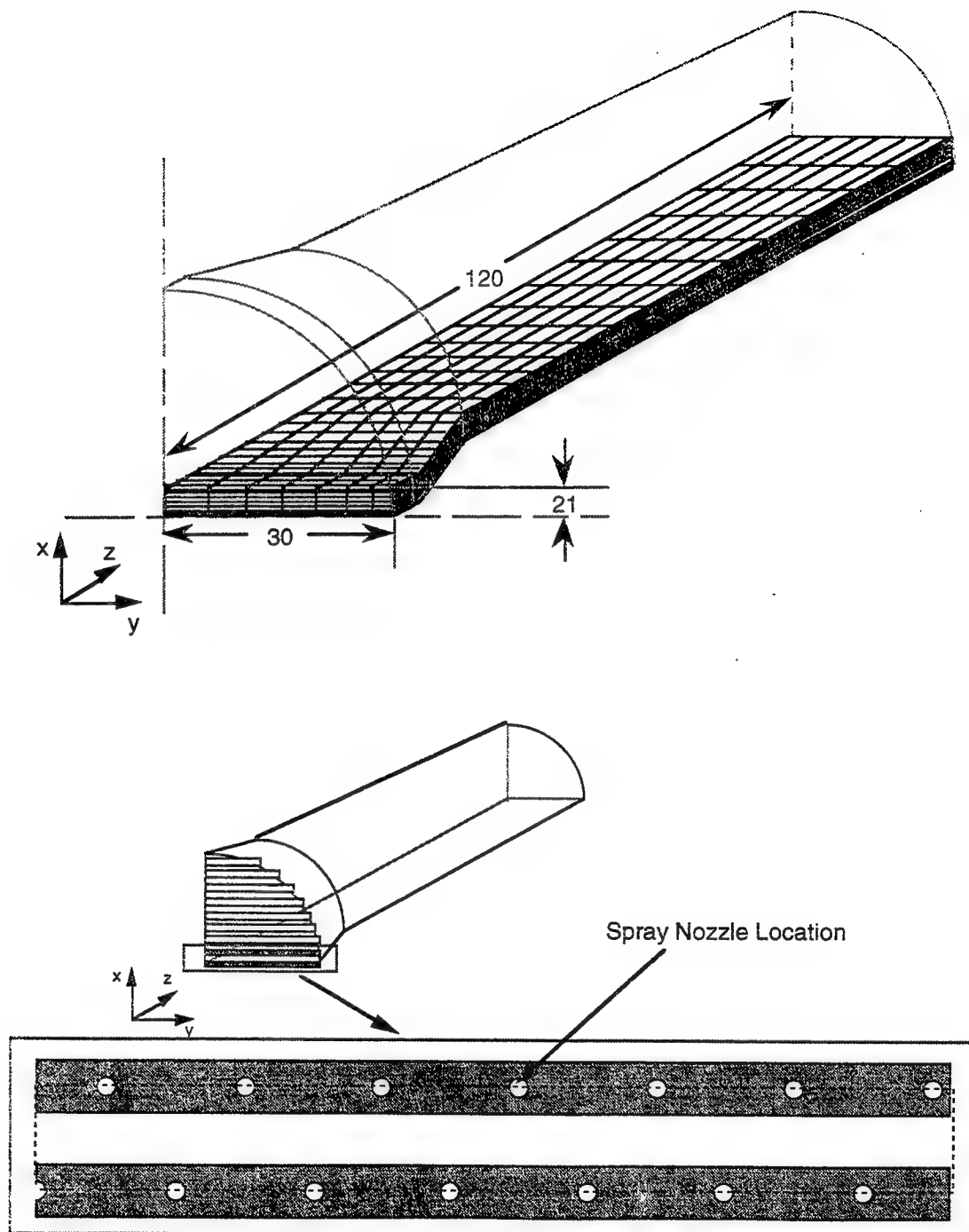


Fig. 1 ASTF quarter section with extracted three-dimensional grid

were treated in the same manner, so that droplets were allowed to bounce elastically when impacting these walls. Note that the extracted flow field includes thirteen half-nozzles and a single quarter-nozzle in the corner. Much of this work and the subsequent parametric analyses were performed at UIUC prior to this summer.

The final portion of this investigation involved the study of the effect of vapor pressure on LWC at the exit plane of the domain. It was theorized that due to the sub-freezing temperatures involved in icing applications, evaporation would be minimal and the resulting vapor pressure could be considered zero. If this were the case, the vapor convection and diffusion calculations could be eliminated, thus creating large savings in both computational time and memory. Spatially-averaged LWC, or $\overline{\text{LWC}}$, and LWC uniformity were measured for the case of accumulated vapor and for the case of zero vapor pressure. The first case involved the evaporation of droplets along with the convection and diffusion of the generated water vapor. The second case involved only the evaporation calculations, because the generated vapor is simply removed from the flow field by setting the vapor density (and thus vapor pressure) to zero.

Comparisons were completed using three different droplet conditions. The first condition is the baseline; second is a decreased droplet injection speed; and the final condition is an increased droplet injection Sauter mean radius. The results are shown in Table 1. Note that $\overline{\text{LWC}}$ is normalized by the spatially-averaged LWC calculated for the baseline case with no evaporation. Therefore, the liquid mass loss in the baseline condition due to evaporation is approximately 9%. Also, σ is the standard deviation of the spatial distribution of LWC when LWC is measured in 210 equally-sized mesh squares at the exit plane. σ is a measure of the LWC uniformity, the more critical icing tunnel parameter -- the lower the standard deviation, the more uniform the LWC. As can be seen, the differences for both the $\overline{\text{LWC}}$ and σ are negligible when comparing vapor versus no vapor pressure. The benefits are illustrated by comparing the CPU requirements. Consistently, necessary computational time decreases by a factor of 1/3 when vapor calculations are eliminated. In addition, memory requirements are reduced by an order of 1/2.

Table 1 Results of evaporation parameter studies.

Condition	\overline{LWC} with vapor	\overline{LWC} no vapor pressure	σ with vapor	σ no vapor pressure	CPU* with vapor	CPU* no vapor pressure
Baseline	0.916	0.914	0.180	0.183	2.20 hr	0.79 hr
$V_{PI} = V_{PI\ BL} - 25\ m/s$	0.896	0.902	0.218	0.215	2.18 hr	0.78 hr
$r_{32} = r_{32\ BL} + 15\ \mu m$	0.981	0.980	0.213	0.214	2.25 hr	0.81 hr

* CPU time based on a Cray C-90 operating at ~ 300 Mflops

The main conclusion to be drawn from these studies is that tracking water vapor is not necessary in these types of calculations. Since the most critical parameters, \overline{LWC} and σ , are nearly identical whether or not vapor calculations are performed, the implication is that the minimal vapor pressure generated in icing tunnel operations has little effect on the evaporation or dispersion of the droplets. Thus, accurate computations can be made at significantly reduced computational cost.

Development and Testing of Acceleration Schemes

Originally, it was proposed by that two different methods could be employed to improve computational efficiency of the droplet calculations of icing simulations (personal communication with Eric Loth). The first is a local acceleration of the vapor convection computations, an Eulerian acceleration. The second method is a novel acceleration scheme of the droplet phase, a Lagrangian acceleration. Both schemes were tested by DeAngelis (1994) in nominally 2-D flow fields with varying success. With the results of parametric studies described above, the application of the Eulerian acceleration scheme was eliminated based on the removal of all the vapor calculations. Lagrangian acceleration modifications were, however, retained and tested for performance with a 3-D flow field.

The theory behind the Lagrangian acceleration scheme is based on the premise of flux equivalency. KIVA-II uses a stochastic eddy model for the turbulent dispersion of particles. As a particle enters an eddy, a turbulent gas velocity is sampled from a Gaussian distribution. The particle experiences this turbulent velocity for an eddy interaction time, Δt_{turbt} , defined by the minimum of either the eddy breakup time or the time it would take the particle to travel across the eddy. To accelerate the convergence of the droplet fluxes, each particle is assigned an individual timestep, Δt_p , which is equal to the maximum of either the global timestep (the timestep that would be used without acceleration) or the current eddy interaction time. With the individual particle timestep, the particle position is updated by the equation

$$\bar{x}_p^{n+1} = \bar{x}_p^n + \bar{v}_p^{n+1} \Delta t_p^{n+1} \quad (1)$$

With larger timesteps, the particles are allowed to reach a converged flux at an accelerated rate. Time accuracy is lost, but the flux of particles through a face per timestep will converge to the correct value and at a reduced number of timesteps (DeAngelis, 1994). The computational savings are two-fold. First, the particle fluxes reach steady state quicker so that less cycles are required for convergence. Second, the number of particles in a given domain at any particular time is less because the particles are traveling out of the domain at a faster rate, providing smaller particle arrays to loop over.

A certain degree of accuracy is theoretically lost due to the increasing size of the truncation error term dropped from Eq. 1:

$$\text{TE} = \frac{1}{2} \frac{Dv_p^{n+1}}{Dt} (\Delta t_p^{n+1})^2 \quad (2)$$

A limiter was therefore placed on the size of the particle timestep so that

$$\Delta t'_p = \max(\Delta t_{\text{global}}, \Delta t_{\text{lim}}) \quad (3)$$

where Δt_{lim} is the minimum of either Δt_{turbl} , or the timestep that would allow the particle to travel more than f_a number of computational cells.

The same flow field used in the parametric studies was used to test the performance of the Lagrangian acceleration scheme. The parameter σ was used to test the accuracy of the calculations over a range of values for f_a as compared to a baseline case with $\Delta t_{\text{global}} = 1.2 \times 10^{-4}$ sec. Results from these tests are shown in Fig. 2 along with the required computational time for the same number of timesteps (thus the savings from convergence in fewer timesteps are not illustrated). This plot shows that there is an extremely limited range of f_a that produces accurate results as compared to the baseline. Though maximum errors are on the order of 9%, this was large in comparison to the required sensitivity of σ as determined by parametric studies when σ variations of only 6% were noted as significant (DeAngelis, 1995). Due to the required level of accuracy with these types of calculations, the Lagrangian acceleration scheme was removed from the current methodology.

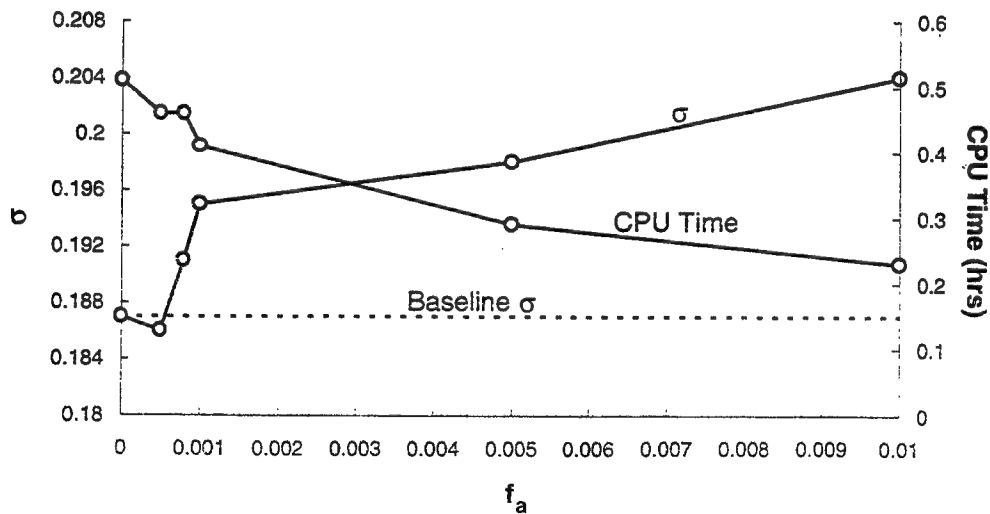


Fig. 2 LWC uniformity and CPU time vs. Lagrangian acceleration timestep limiter

Validation of Current Methodology

Two different experimental databases were considered for validation cases of the current NPARC/modified KIVA-II methodology. The first, the R-1D database, involved a flow through a plenum, bellmouth and test section. The plenum housed a spray nozzle, similar to the nozzles of the ASTF wind tunnel, injecting water downstream through the bellmouth and test section. The second test case, the Snyder and Lumley database, is a decaying turbulence channel flow with particles of different types injected isokinetically along the centerline of the flow.

R-1D Test Case

Tests generated in AEDC test cell R-1D were used as a database for validation of particle trajectory calculations. Previously, this same database was used for validation of the FASERVT particle trajectory code created by Lankford (1986). A sketch of the experimental apparatus is shown in Fig. 3. All tunnel dimensions were provided by Scott Bartett. A nozzle held at the centerline of a 3 ft inner diameter plenum sprays water downstream through a bellmouth, pipe and test section. Air flow in the plenum at a total pressure of 14.2 psia was accelerated through the bellmouth to two different static pressures for two separate test section flow speeds. Only one static pressure, 13.8 psia, was considered here. This generates a test section velocity of approximately 230 ft/s. LWC was measured in the near-field of the nozzle (position A) and in the far-field (position B).

Calculations of the air flow field were performed using NPARC. A grid of approximately 300,000 points for a half-cylinder symmetric domain was generated, and all physical surfaces were defined as slip walls. The downstream exhaust duct was eliminated and the pipe was lengthened to include the measurement location B. This was done to decrease the grid matrix size and was possible because, experimentally, air was bled into the duct upstream of the pipe exit to attempt to reduce recirculation. The Chien $k-\epsilon$ model was employed to calculate turbulence with a prescribed freestream turbulent intensity of 7%. At the time of this writing, the residuals had not fallen to a sufficient level and the calculations continue.

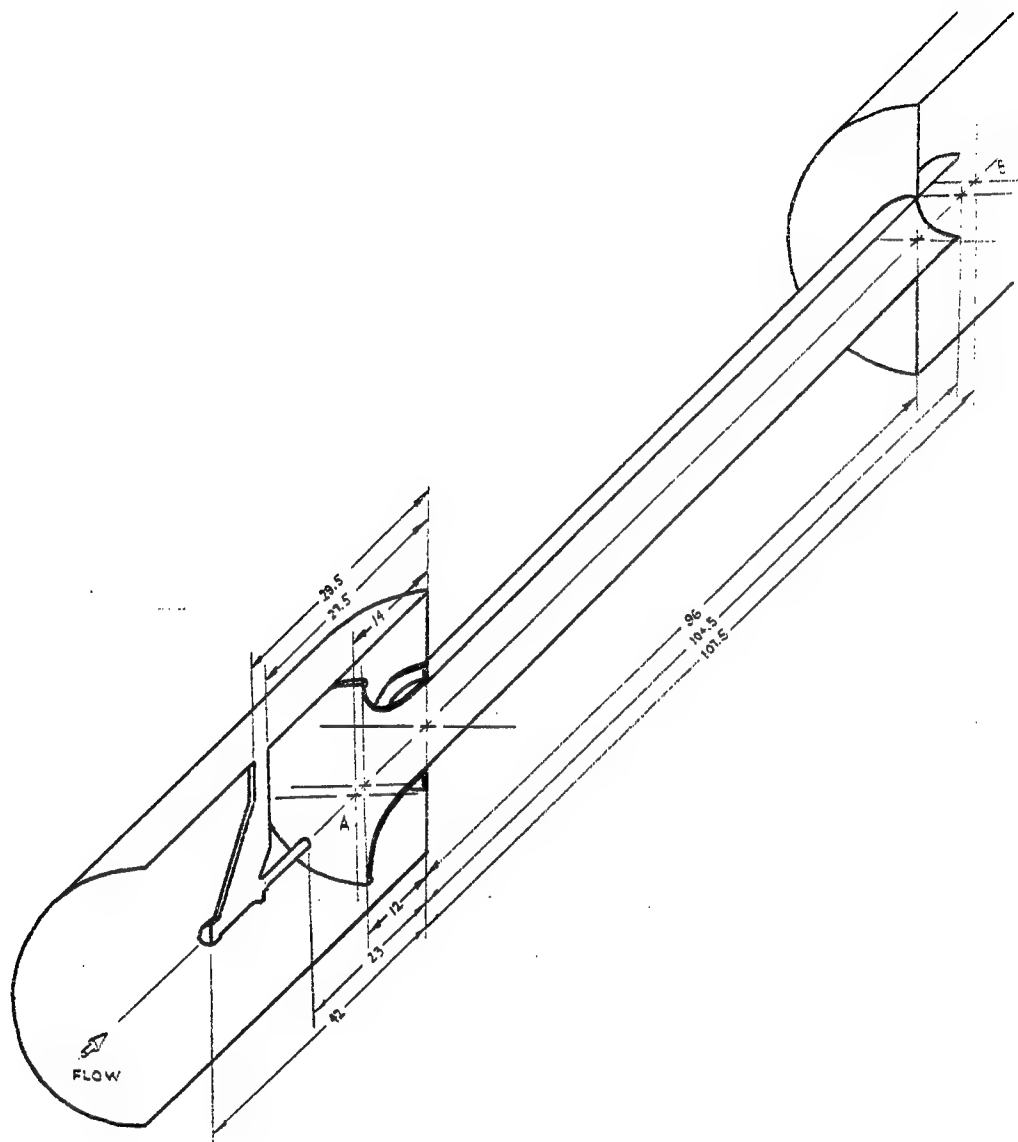


Fig. 3 R-1D test cell configuration

Snyder and Lumley Test Case

Snyder and Lumley (1971) performed experimental tests of particle dispersion in a 16 ft x 16 inch x 16 inch vertical wind tunnel. The air flow was low speed and turbulence was generated by the presence of a grid of mesh size 2.54 cm at an upstream portion of the tunnel ($x = 0$). All flow variables, including k and ϵ , were measured and expressed as functions of the tunnel axial position. The data set has been used successfully in many particle code validation cases including Gosman and Ioannides (1981) and Shuen *et al.* (1983).

A grid and flow field were input into the modified KIVA-II. The grid was resolved in the axial direction by 100 uniform cells from the position $x = 46.8$ cm to $x = 446.8$ cm. There were no variations of conditions in the y or z directions. The values of the flow variables assigned to appropriate grid points are shown in Fig. 4, however, the turbulence quantities k and ϵ were assigned directly based on the particle x position to eliminate the influence of the grid spacing. Four types of particles, as in the experiment, were injected along the centerline with $U_p \approx 655$ cm/s at $x = 50.8$ cm. The characteristics of the four particle types are shown in Table 2. Dispersion was measured in the same manner as Snyder and Lumley; i.e. after particles crossed $x = 172.72$ cm ($t = 0$), their displacement Y from their y position at $t = 0$ was measured. Note that dispersion in the y or z directions are statistically equal due to symmetry. A statistically sufficient 800 particles were used in each case as compared to approximately 700 particles used experimentally, which provided less than 10% error.

Table 2 Particle descriptions for Snyder and Lumley experiment.

Particle Type	Diameter (μm)	Density (g/cm^3)
Hollow glass	46.5	0.26
Solid glass	87.0	2.5
Corn Pollen	87.0	1.0
Copper	46.5	8.9

$$\begin{aligned}
 u &= \bar{U} \quad v = w = 0 \\
 p &= 1 \text{ atm} \\
 T &= 299 \text{ K} \\
 k &= \frac{1}{2} \bar{U}^2 \left[\frac{1}{42.4 \left(\frac{x}{M} - 16 \right)} + \frac{2}{39.4 \left(\frac{x}{M} - 12 \right)} \right] \\
 \varepsilon &= \frac{1}{2} \bar{U}^3 \left[\frac{1}{42.4 M \left(\frac{x}{M} - 16 \right)^2} + \frac{2}{39.4 M \left(\frac{x}{M} - 12 \right)^2} \right] \\
 \bar{U} &= 655 \text{ cm/s} \quad M = 2.54 \text{ cm}
 \end{aligned}$$

Fig. 4 Flow expressions for Snyder and Lumley experiment

Extremely large errors in Y were discovered when using the definitions for the particle-eddy interaction time (Δt_{turbt}) provided by KIVA-II. Shuen *et al.* (1983) note that due to the somewhat arbitrary definitions of eddy length and time scales, the stochastic model, which is very similar to the KIVA-II model, must be calibrated. For this case, modifying the eddy breakup time, t_e , and the eddy traversal time, t_t , increased the accuracy. In both cases, the time scales were approximately 4 times larger than what was accurately needed. Their respective values were equivalent to:

$$t_{e \text{ new}} = 0.24 t_{e \text{ KIVA-II}}$$

and

$$t_{t \text{ new}} = 0.26 t_{t \text{ KIVA-II}}$$

The results for the four particles in terms of the mean square of Y are shown in Fig. 5. Across the range of particle densities and sizes, dispersion is captured well with the consistent definitions for t_e and t_t . As noted by Shuen and confirmed here, the dispersion of the lighter particles (hollow glass and corn pollen) was dictated primarily by the eddy breakup time, and the dispersion of the heavier particles (solid glass and copper) was controlled mainly by the eddy traversal time. Also shown in Fig. 5 by the broken lines are two curves generated for hollow glass and copper using the original KIVA-II definitions of t_e and t_t illustrating the large errors in dispersion.

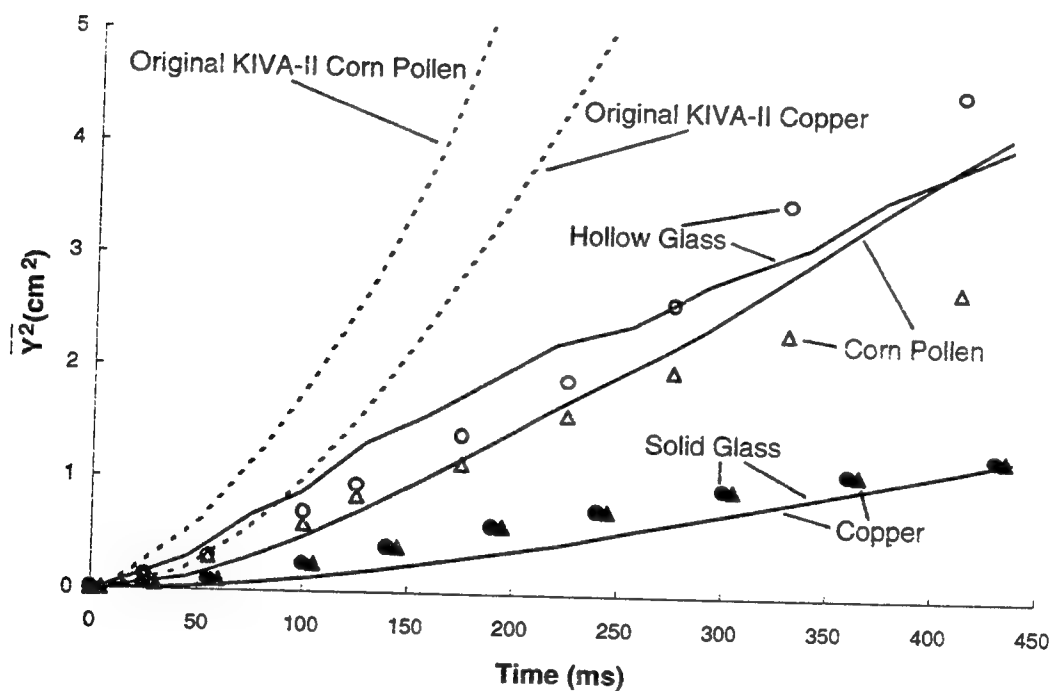


Fig. 5 Particle dispersion vs time measured since particles crossed $x = 172.72$ cm (computations are indicated by solid lines; experimental data are indicated by data points; computations previous to calibrations are indicated by broken lines)

Creation of a Unified Icing Code

AEDC desired a single icing test simulation code in a workstation version that would include all modifications made to date by AEDC and UIUC. The modified version of KIVA-II was used as a basis for developing this unified code, tentatively named K-ICE, as an SGI version. Pertinent additions and improvements to the modified KIVA-II are noted here:

1. The evaporation subroutine was modified to account for constant zero-vapor pressure and all vapor calculation subroutines were removed.
2. Additions to include rotating flow capability were incorporated into K-ICE. Dennis Lankford had previously included terms accounting for centrifugal and Coriolis acceleration of particles into the equations of motion. The use of these terms was made optional by computational block.
3. Two particle arrays were added to assign two additional particle characteristics. The first assigns the number of the nozzle from which the particle was injected. The second assigns a unique number to a particle at injection (based on the total number of particles already injected) to allow for particle tracking even when particle arrays are re-packed after destruction or shattering.
4. Nozzles can be defined as partial nozzles if they are placed along a symmetry plane.
5. Inputting of parameters was modified into a namelist format.
6. Cell flagging for boundary conditions has been reconfigured for robustness.
7. Output is now made in three possible ways: instantaneous particle information, particle tracing information and accumulation plane information.

A K-ICE user's manual has been prepared. The manual takes the user through the inputting and execution requirements followed by an explanation of the output format. All input variables are defined in detail and boundary conditions are explained in depth.

Conclusions and Future Studies

The current methodology has been finalized by activities this summer. For applications to icing simulations, K-ICE performs much more efficiently than previous KIVA-II versions. This has been accomplished by reducing the algorithm to a series of one-dimensional array loops over the particles now that evaporation and vapor calculations have been eliminated. The computational architecture allows for convenient and timely parametric studies of a single air flow condition in a wind tunnel or in rotating turbomachinery.

The validation and subsequent calibration of the stochastic eddy model, however, indicate the need for more investigation into particle dispersion simulations. The arbitrary definitions of the eddy scales are correct only when calibrated. It is possible that for different flow regimes, different calibration constants may be necessary. This will become evident when validation of the R-1D test is completed.

Continuation of the R-1D validation is crucial. By testing the data of Snyder and Lumley, only the particle dispersion model is validated, and not necessarily the entire methodology. The R-1D database will also allow for more extensive parametric studies, including the investigation of the influence of the nozzle initial conditions.

References

- Amsden, A.A., P.J. O'Rourke and T.D. Butler, (1989) "KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays" DE89-012805, Los Alamos National Laboratory, Los Alamos, NM.
- Barlett, C.S. and W.J. Phares (1993) "Icing Testing of a Large Full-Scale Inlet at the Arnold Engineering Development Center" AIAA-93-0299.
- Cooper, G.K. and J.R. Sirbaugh (1989) "PARC Code: Theory and Usage" AEDC TR 89-15, Arnold Air Force Base, TN.
- DeAngelis, B.C. (1994) "Performance and Validation Studies of the KIVA-II Code for Wind Tunnel Spray Bar Droplet Dispersion" *Final Report for AFOSR Summer Graduate Student Research Program, AEDC.*
- DeAngelis, B.C. (1995) *Notebook for AEDC Summer Graduate Student Research Program.*
- Gosman, A.D. and E. Ioannides (1981) "Aspects of Computer Simulation of Liquid-Fueled Combustors" AIAA-81-0323.
- Lankford, D.W. (1986) "Icing Distribution Code, FASERVT" *AEDC User's Manual.*
- Loth, E. (1993) "Computational Methodology for Wind Tunnel Spray Bar Droplet Dispersion" *Final Report for AFOSR Summer Faculty Research Program, AEDC.*

Shuen, J.S., L.D. Chen and G.M. Faeth (1983) "Evaluation of a Stochastic Model of Particle Dispersion in a Turbulent Round Jet" *AIChE Journal*, **29**, pp. 167-170.

Snyder, W.H. and J.L. Lumley (1971) "Some Measurements of Particle Velocity Autocorrelation Functions in a Turbulent Flow" *Journal of Fluid Mechanics*, **48**, pp. 41-71.

PREDICTION OF COOLING WALL TEMPERATURES USING THE NPARC CODE

David Dembicki
Graduate Research Assistant
Department of Mechanical Engineering

University of Tennessee Space Institute
Tullahoma, TN

Final Report for:
Graduate Student Research Program

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and
Arnold Engineering Development Center

August 1995

PREDICTION OF COOLING WALL TEMPERATURES USING THE NPARC CODE

David Dembicki
Graduate Research Assistant
Department of Mechanical Engineering
University of Tennessee Space Institute

Abstract

Research was conducted in the area of predicting thermal effects within a F100 class turbine engine afterburning section using a Navier-Stokes flow solver set up for a 2D, axi-symmetric case. Particular attention was paid to the simulation of the engine's flame holders and porous cooling liner. The complex flame holder geometry was portrayed as an equivalent two-dimensional blockage area. The liner was modeled as both a porous slip wall and as a solid no-slip wall. While the various shapes used for the flame holders appeared to have some influence on the cooling liner temperatures, the type of wall used for the liner had a major effect on the final wall temperatures. The porous slip wall produced unfavorable results; however, the no-slip, solid wall yielded results similar to the experimental data. Flame holder location also played a significant part in achieving the correct liner wall temperatures.

PREDICTION OF COOLING WALL TEMPERATURES USING THE NPARC CODE

David Dembicki

Introduction

Since its inception, computational fluid dynamics (CFD) has had enormous impact on the aerospace community. CFD allows the engineer to examine numerous design options without actually taking the time to build, configure, and re-configure experimental set-ups. CFD can also be used to model an already existing design. This ability of computational fluid dynamics has been widely exploited in the realm of jet engine testing. An area of interest for the Arnold Engineering Development Center is the accuracy with which CFD can predict the temperatures of an engine cooling liner operating under flight conditions.

Methodology

Before conducting the CFD analysis to determine the cooling liner temperatures within a F100 class engine, the engine geometry had to be determined for the test case in question, and the appropriate computational grid generated. The following describes the procedure used to develop a linkage model. The linkage model determines the locations of specific points along the convergent and divergent sections of the engine nozzle given the nozzle throat area and the exit area. The linkage model also assumes that the engine structure is essentially rigid. Figure 1 shows the generic layout of the nozzle portion of a jet engine.

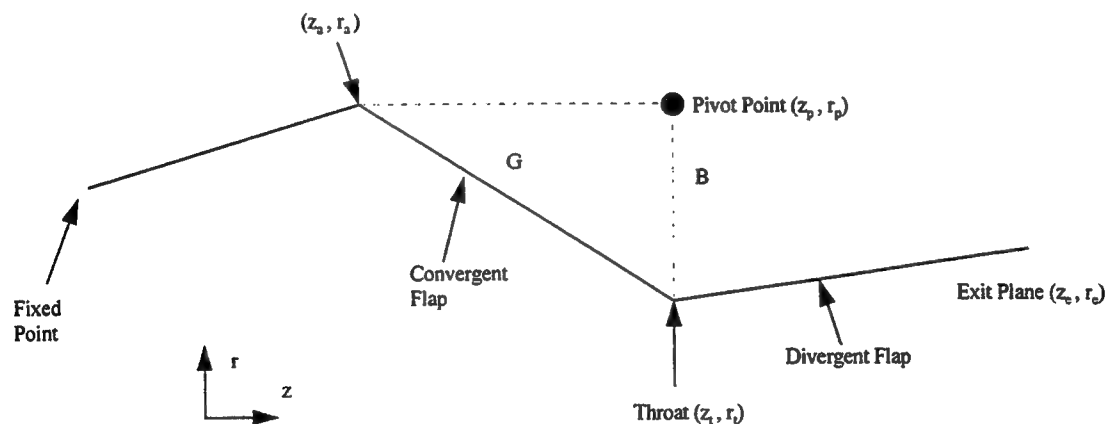


Figure 1. Jet Engine Nozzle

Assuming that the initial coordinates and the areas for the throat and exit plane are given, begin as follows:

Given: $(z_a, r_a); (z_p, r_p); (z_t, r_t); (z_e, r_e)$ initial coordinates

$A_t; A_e$

throat and exit areas at the desired conditions

Find:

$(z_a', r_a'); (z_t', r_t'); (z_e', r_e')$

the corresponding new positions

From geometry:

$$G = \left[(z_t - z_a)^2 + (r_t - r_a)^2 \right]^{1/2} = \text{constant}$$

$$B = \left[(z_t - z_p)^2 + (r_t - r_p)^2 \right]^{1/2} = \text{constant}$$

$$r_t' = \left(\frac{A_t}{p} \right)^{1/2}$$

$$z_t' = z_p + \left[B^2 - (r_t' - r_p)^2 \right]^{1/2}$$

$z_p + \dots$ chosen from geometry considerations.

Now consider Figure 2:

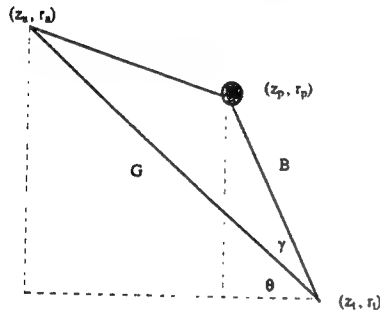


Figure 2. Nozzle Converging Section

From this figure the following equations can be derived:

$$\theta = \sin^{-1} \left(\frac{r_a - r_t}{G} \right) \quad \beta = \sin^{-1} \left(\frac{r_p - r_t}{B} \right) \quad \gamma = \beta - \theta$$

Consider:

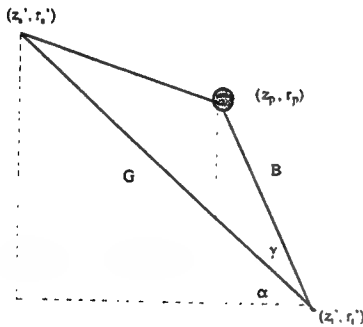


Figure 3. Nozzle Converging Section

$$\sin(\alpha + \gamma) = \frac{r_p - r_t'}{B}$$

$$\alpha = \sin^{-1} \left(\frac{r_p - r_t'}{B} \right) - \gamma$$

$$\sin \alpha = \frac{r_a' - r_t'}{G}$$

$$r_a' = r_t' + G \sin \alpha$$

$$\cos \alpha = \frac{z_t' - z_a'}{G}$$

$$z_a' = z_t' - G \cos \alpha$$

Finally consider Figure 4:

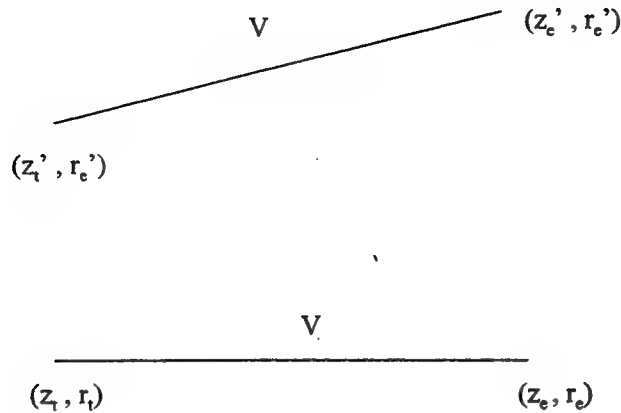


Figure 4. Nozzle Diverging Section

$$V = \left[(z_e - z_t)^2 + (r_e - r_t)^2 \right]^{1/2} = \left[(z_e' - z_t')^2 + (r_e' - r_t')^2 \right]^{1/2} = \text{constant}$$

$$r_e' = \left(\frac{A_e}{\pi} \right)^{1/2}$$

Therefore, solving the V equation for z_e' yields the following:

$$z_e' = z_t' + \left[V^2 - (r_e' - r_t')^2 \right]^{1/2} \quad z_t' + \dots \text{ is selected due to geometry considerations.}$$

The equations for r_e' and z_e' then give the positions of the points needed to create the engine geometry. In summary, the linkage model is established through geometric relations. Knowing the initial positions along with the throat and exit areas at the desired conditions, the locations of the convergent and divergent nozzle sections can be determined.

After the engine geometry was determined, GRIDGEN, a grid generator, was used to create various multi-block grids for the test cases supplied to the flow solver. Also, the boundary conditions for the surfaces of the engine were determined. Figure 5 is a representation of the engine geometry. The numbers correspond to the boundary conditions applied to each surface. Table 1 describes the various boundary conditions used.

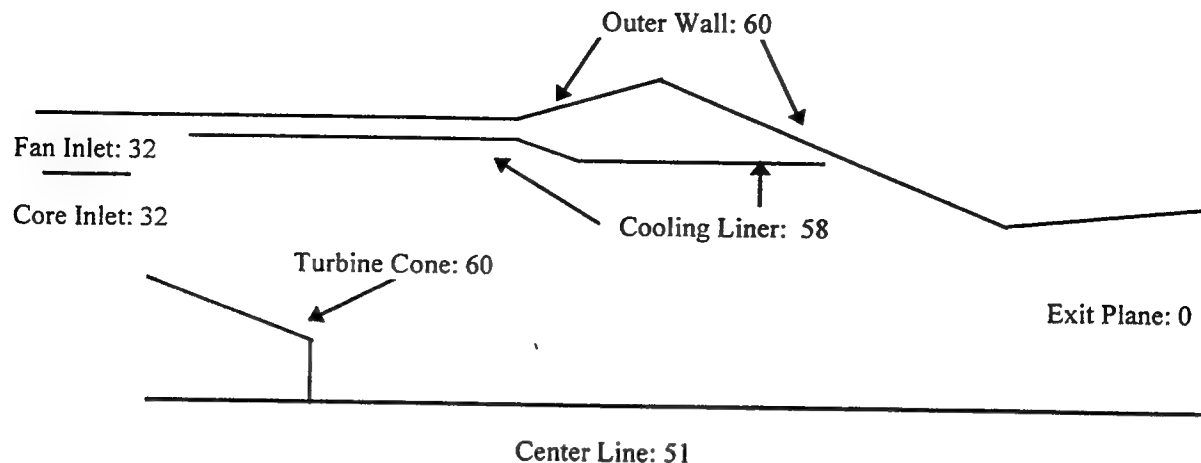


Figure 5. Cross-section of Engine Geometry

Table 1. NPARC Boundary Conditions

Boundary Condition	Description
0	Outflow
32	Variable Total Pressure, Flow Angle, Total Temperature
51	Axis of Symmetry
58	Inviscid Screen Boundary
60	No-slip, Adiabatic Surface

In the above Figure 5, no flame holders were shown for clarity; however, the type 60 (no-slip, adiabatic surface) was applied to the flame holders. The location of the flame holders is in the region above the turbine cone. The cooling liner boundary condition was varied in all the test cases. The liner was modeled first as a porous, no-slip wall with heat conduction. Then it was represented by a porous, slip wall boundary condition. Finally, a no-slip, adiabatic wall b.c. was applied to the cooling liner wall. Each boundary condition is discussed in the results section.

The variable pressure and temperature inlet profiles were created to reflect the experimentally measured profiles. Figure 6 shows the variable pressure profile obtained, and Figure 7 shows the variable temperature profile obtained. Since no experimental temperature data was available for the bypass duct, a constant temperature profile was used.

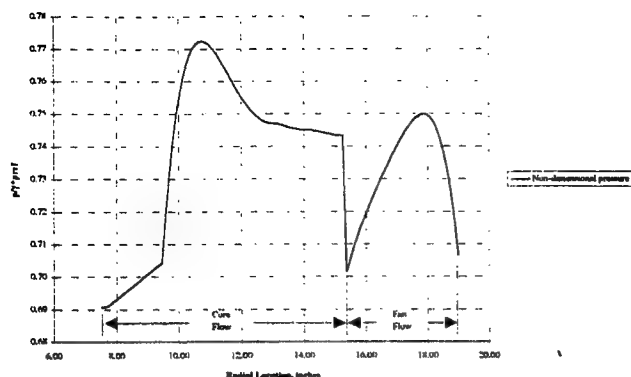


Figure 6. Inlet Non-dimensional Pressure Profile

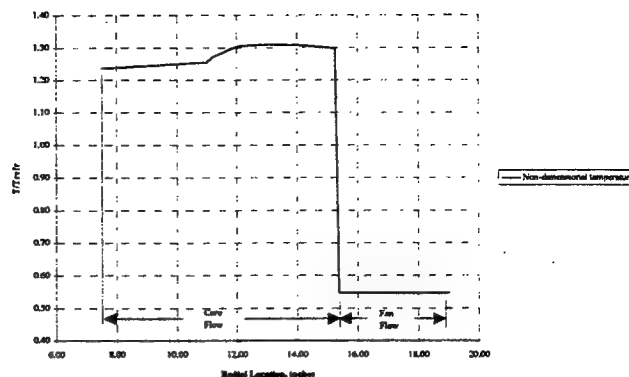


Figure 7. Inlet Non-dimensional Temperature Profile

RESULTS

With the boundary conditions established, the test cases were conducted for flight conditions at Mach number equal to 0.5 and altitude of 10,000 feet. Four individual test cases were performed. While each case incorporated information gained from previous trials, the test cases were treated as single entities designed to achieve the desired cooling liner temperatures. There wasn't just one test case with multiple variations; different approaches were attempted in the various cases. For example, the total blockage area representing the flame holders was held constant, but the location of the blockage changed slightly from test case to test case.

The first test case contained no flame holder blockage area. A porous, no-slip wall with heat conduction (boundary condition type 68) was applied to represent the cooling liner. This boundary condition caused the solution to become unstable quickly. To achieve stability, the porosity of the wall was varied along with the time step of the NPARC code. No-slip, solid walls were also added to the liner. Since none of these attempts stabilized the code, the type 68 boundary condition was replaced by the type 58 boundary condition (Porous, slip wall).

Using the porous, slip-wall boundary condition, the NPARC code was run for 60,000 iterations. Figures 8 and 9 show that the convergence for this initial test case occurred within this number of iterations. The exit plane Mach number shown in Figure 8 matched closely with the value of $M=1.41$, which was obtained by a simple 1D calculation based on the information from the experiment.

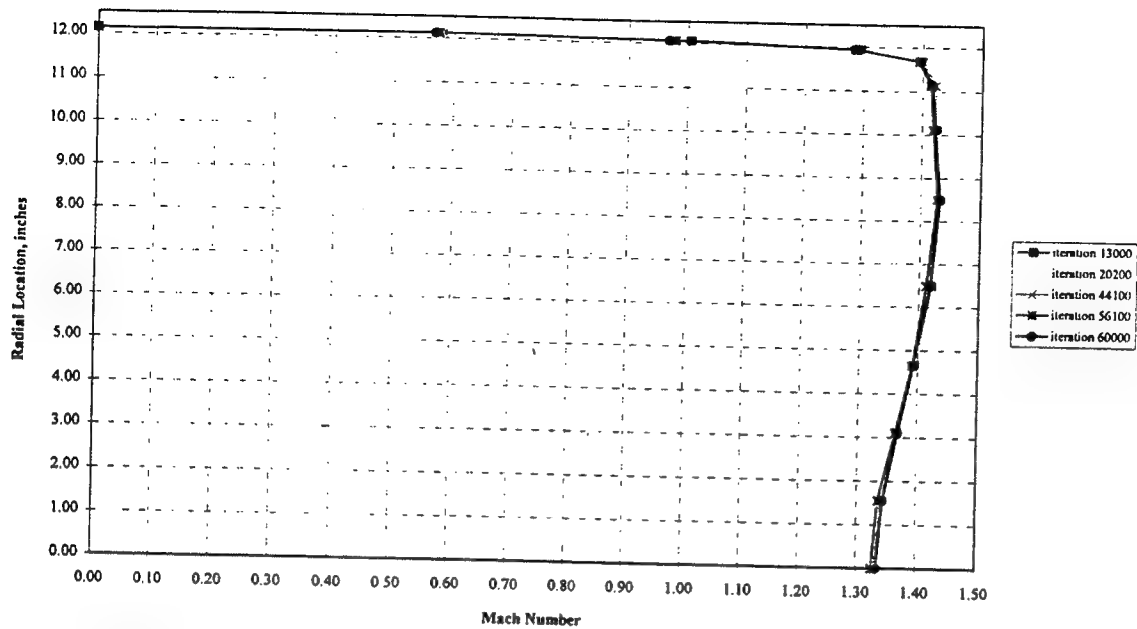


Figure 8. Mach Number Convergence For The Calibration Case #1

Figure 9 shows the wall temperature convergence plot. Here, it is seen that the cooling liner wall temperature reaches value of approximately 0.54 along its entire length. As the next plot illustrates, this is a poor prediction of the actual wall temperatures reached along the cooling liner.

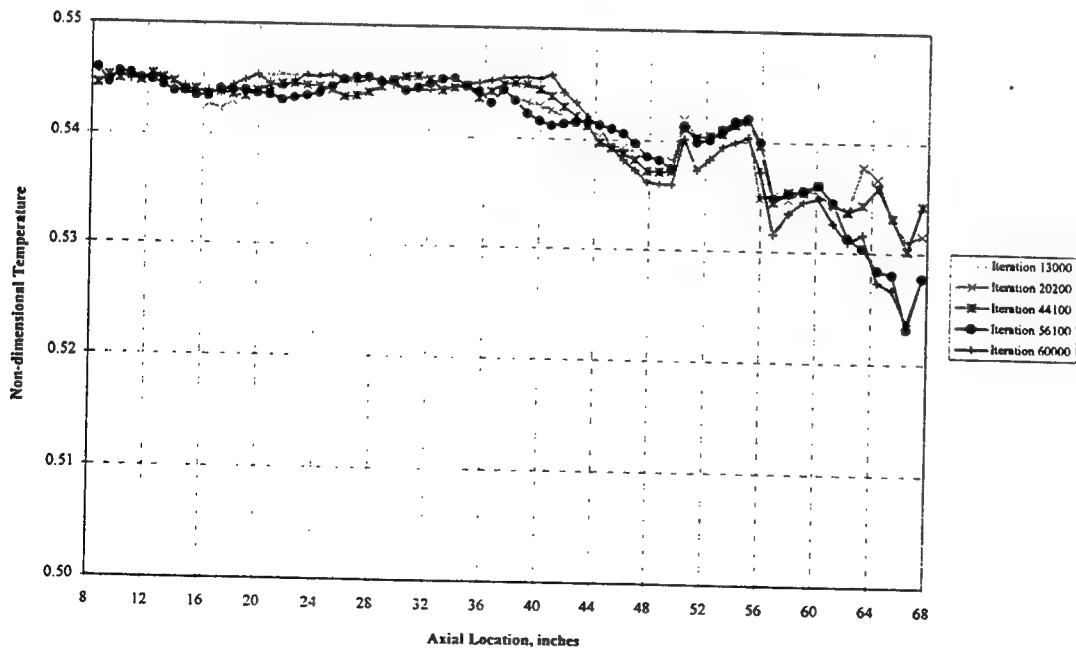


Figure 9. Wall Temperature Convergence For The Calibration Case #1

Looking at Figure 10, it is seen that the temperatures of the liner wall predicted by the NPARC code differed significantly from the experimental data. The largest error between the predicted values and the measured values is 15.2%. Using FAST for flow visualization showed that the fan flow and the core flow failed to mix substantially. This failure of the gas streams to mix implied that the flame holders were a crucial part of the flow mixing in the engine's afterburning section.

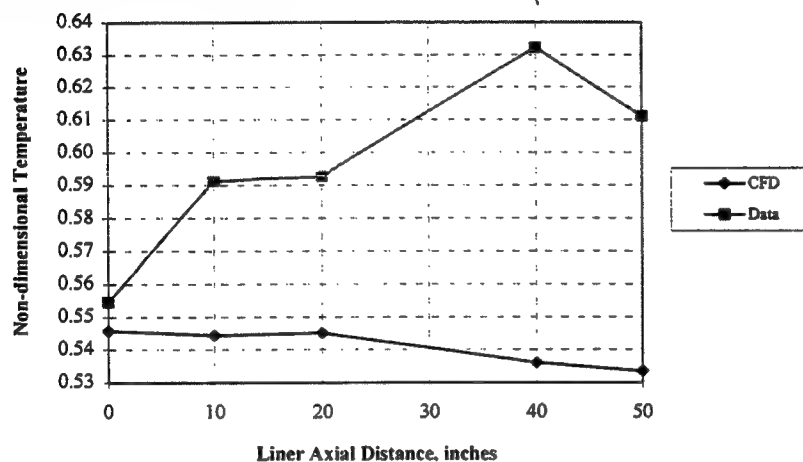


Figure 10. Temperature Comparison For The Calibration Case #1

The second test case included flame holder blockage. It was decided to use a series of radial hoops as a substitute for the full three-dimensional flame holder assembly. To maximize the accuracy of the model, two parameters from the actual flame holder were satisfied. These parameters were: total blockage area and the width of the radial flame holder fingers.

Considering blockage area and the "characteristic" width together, adequate radial hoops were obtained. The following development builds upon work already done by Gary Ledbetter in a study to replace afterburner flame holders with "equivalent hoops"¹. In this study, 8 hoops were established as the "ideal" replacement for the flame holders. Three hoops represent the inner region, 3 hoops represent the outer region, and 2 hoops model the center region where the fingers are attached. The conclusion from Ledbetter yielded the following locations for the blockage areas (hoops) that best model the flame holders and fuel manifolds. Figure 11 shows a graphical representation of the blockage used and Table 2 contains the radial locations for the blockage.



Figure 11. Aft Looking Forward Representation of Radial Blockage Area For Test Case # 2

Table 2. Test Case # 2 Blockage Area Locations

Point	Radial Distance, inches
A	7.57
B	8.07
C	8.84
D	9.34
E	9.819
F	10.969
G	11.570
H	12.643
I	12.695
J	13.680
K	14.05
L	14.43
M	15.74
N	16.12
O	16.329
P	17.694

Points G-J were "collapsed" to represent a single, triangular blockage area. The rest of the flame holder blockage was represented as rectangles placed at the appropriate radial locations. The width of the rectangles was the actual flame holder width.

With the flame holder positions established, the second test case was conducted. In this case the porous, no-slip wall with heat conduction boundary condition again failed to work properly and was replaced by the porous slip-wall boundary condition. When using this porous, slip-wall boundary condition, the cooling liner failed to reach the temperatures given by the experimental data. In fact, the temperatures at the wall were very similar to those computed during the calibration case. The lack of temperature rise along the wall was partially attributed to the fact that the porous, slip-wall boundary condition did not account for the boundary layer effects along the wall.

It was decided to replace the type 58 porous, slip-wall b.c. with an adiabatic, no-slip wall type 60 boundary condition to include the effects of the wall's boundary layer. While using this adiabatic, no-slip wall boundary condition, flow visualization showed that mixing between the fan flow and the core flow was enhanced. Both Figure 12 and 13 indicate that the test case converged in approximately 55,000 iterations. Exact convergence was not achieved due to time constraints and some inherent unsteadiness within the simulated flow.

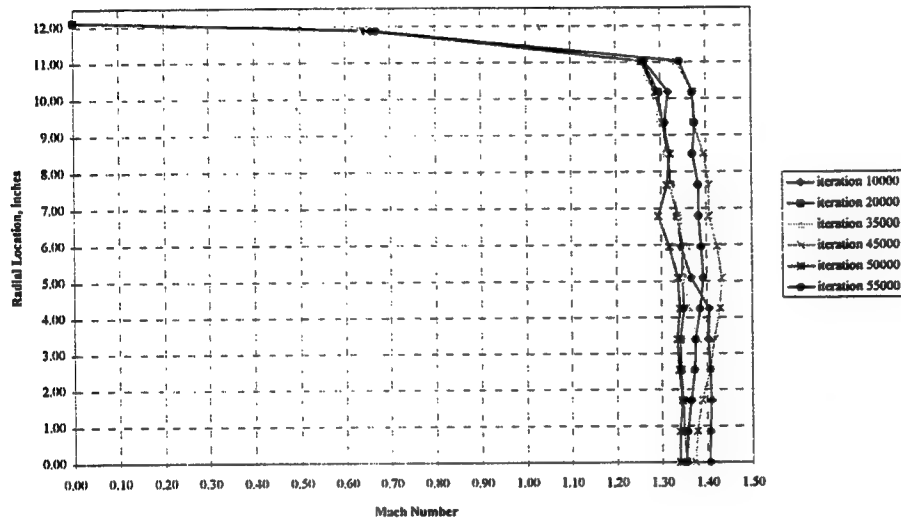


Figure 12. Mach Number Convergence For The Second Test Case

Figure 12 above shows that again the exit plane Mach number agreed favorably with the value attained from the experimental calculation. Figure 13 below illustrates the "near" convergence mentioned previously.

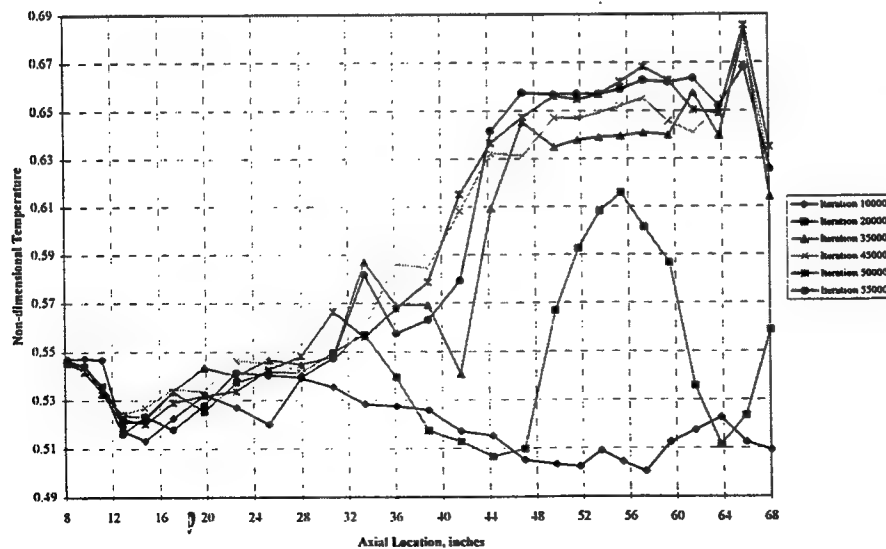


Figure 13. Temperature Convergence For The Second Test Case

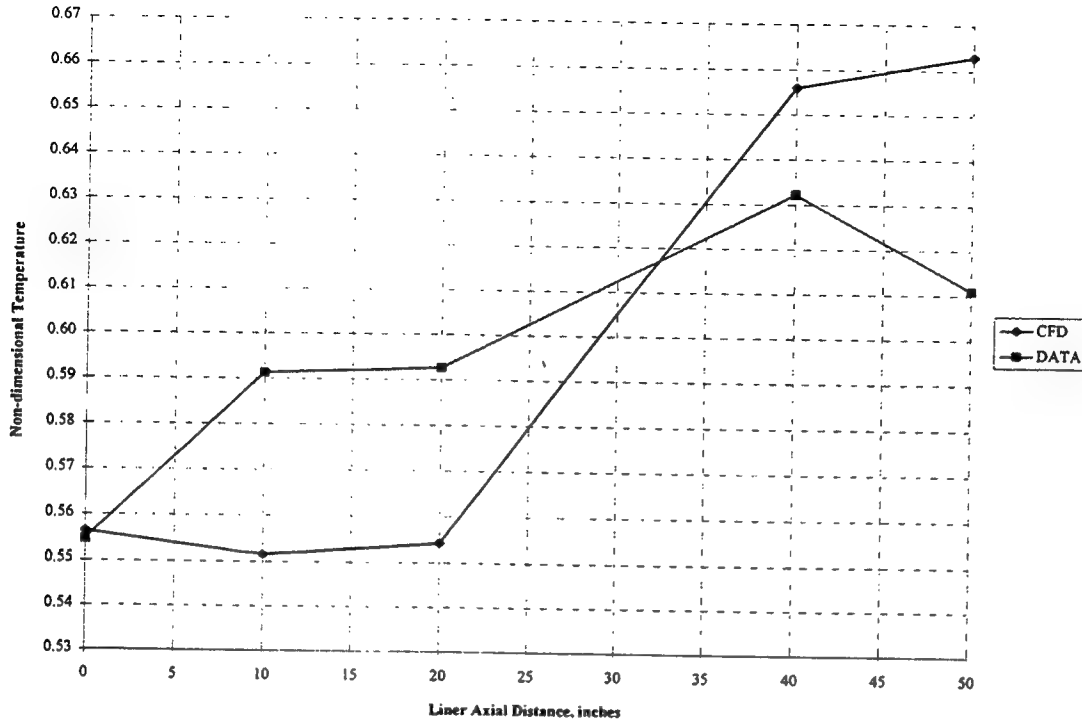


Figure 14. Temperature Comparison For The Second Test Case

Looking at Figure 14, the temperature of the cooling liner's leading edge is almost exactly that given by experiment. In the region of 1 inch to 20 inches from the leading edge, the wall temperatures were off by a maximum of 6.74%. This temperature deficit at the front of the wall was attributed to poor mixing in the region near the flame holder blockage area, an inherent problem in representing a 3D object as a 2D shape. Beginning in the area 30 inches from the leading edge, the cooling liner wall temperatures are more accurately predicted until reaching the axial distance of 50 inches, where the wall temperature was over predicted by 8.50%.

Test case three built upon the information gathered from the second test case. The cooling liner wall was represented by the no-slip, adiabatic wall b.c. when both porous wall boundary conditions proved less than satisfactory. The flame holder blockage area was maintained, yet the shape and location of the blockages were changed. Instead of using the rectangles to represent the flame holders, triangular shapes similar to the cross-section of the flame holders were implemented. Also, the radial locations of the blockage areas were changed to distribute the flame holder area more evenly. Figure 15 shows the flame holder model used, and Table 3 contains the new locations for the flame holder representations.

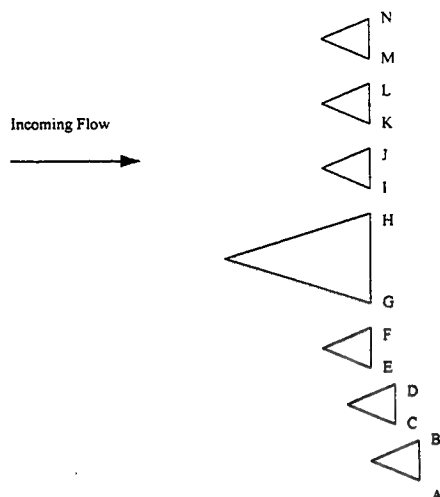


Figure 15. Flame Holder Representation For Test Case 3

Table 3. Test Case 3 Blockage Area Locations

Point	Radial Location, inches
A	7.664
B	8.400
C	8.900
D	9.651
E	10.218
F	10.969
G	11.611
H	13.680
I	14.644
J	15.300
K	15.734
L	16.400
M	16.833
N	17.694

Test case #3 followed the pattern established in the previous two cases. Initially, the cooling liner wall was represented by the no-slip, porous wall with heat conduction boundary condition. When this b.c failed to allow the solution to remain stable, it was replaced with the porous, slip wall boundary condition. The wall temperatures predicted using the porous, slip wall b.c were unsatisfactory, so once again the cooling liner was modeled as a no-slip, adiabatic wall.

Convergence for this case occurred in about 50,000 iterations. Looking at Figure 16, the exit plane Mach number is again in agreement with the calculated value of 1.41. Figure 17 shows the inherent unsteadiness of the flow, implied by the “spike” which occurs between 48 and 64 inches. This temperature jump can be explained by the fluctuations in the flow field that happen constantly and merely were caught at this particular iteration. If the

NPARC code was advanced another couple thousand iterations, it would be expected that this “spike” would either shift its location or disappear entirely.

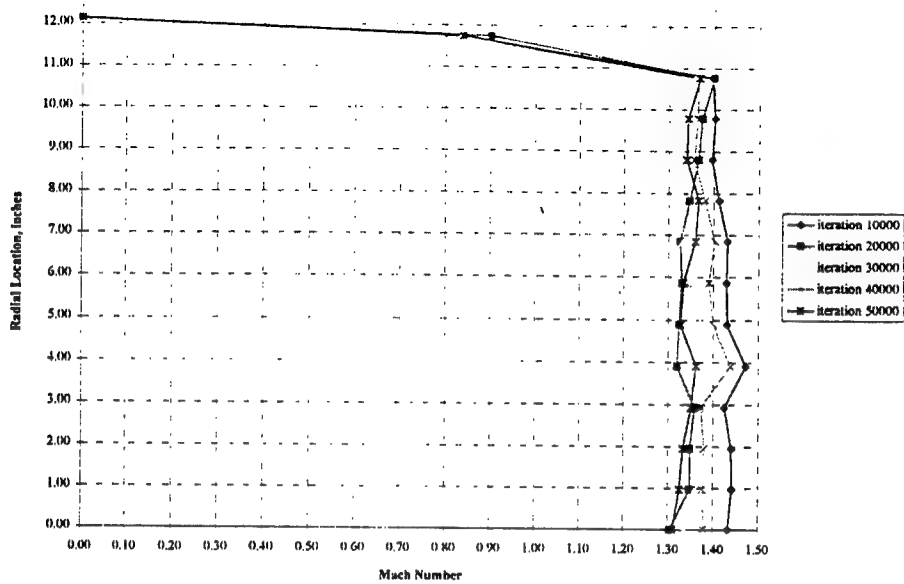


Figure 16. Mach Number Convergence For Test Case #3

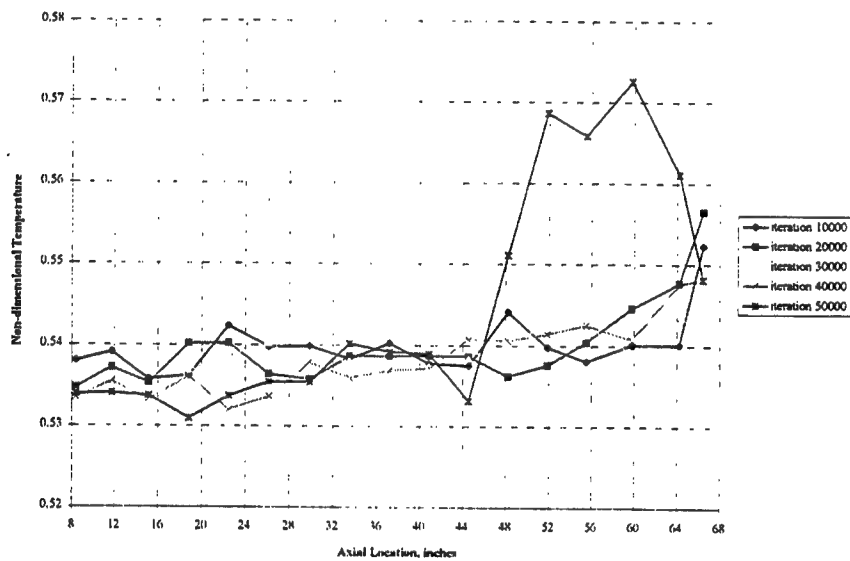


Figure 17. Wall Temperature Convergence For Test Case #3

Figure 18 is a comparison of the wall temperatures predicted by the CFD code and the actual values measured during the experiment. The initial prediction of the cooling liner's leading edge is accurate to within 1%. However, the predicted temperatures for the majority of the wall differ significantly from the experimental data. The

largest error in the prediction of 12.3% comes at the axial location of 40 inches, with the rest of the wall temperatures being under predicted by approximately 8.5%.

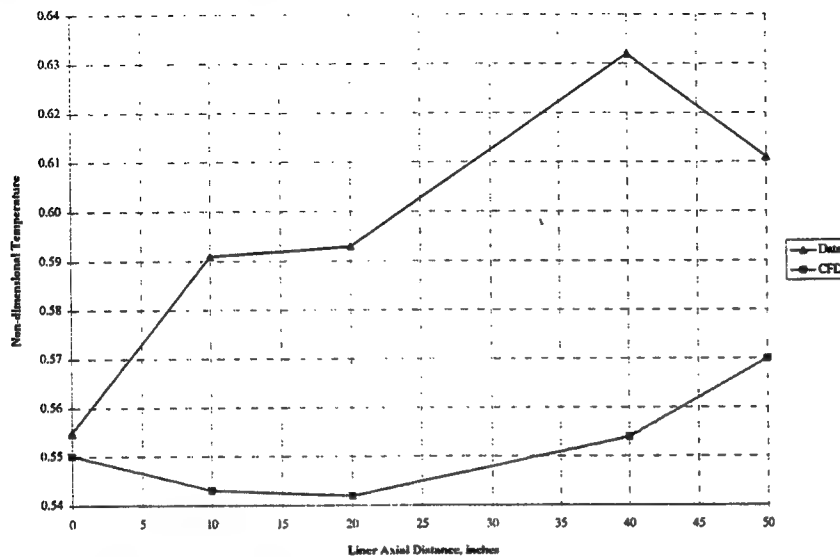


Figure 18. Temperature Comparison For Test Case #3

The fourth and final test case modified both the flame holders' shape and location. This case also changed the shape of the cooling liner wall. In the previous three test cases the liner wall was represented as a straight line. In reality, the cooling liner wall is not straight, it is corrugated or wavy. It was decided to model the liner wall as a wavy line and not a straight line. Again, the liner wall was modeled using the solid wall when neither porous wall b.c. yielded appropriate predictions. Figure 19 shows the shape of the flame holders used in this test case. Table 4 contains the exact locations of the flame holder blockage areas.

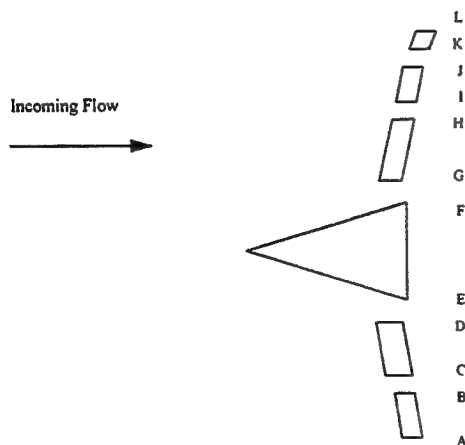


Figure 19. Flame Holder Representation For Test Case 4

Table 4. Test Case 4 Blockage Area Locations

Point	Radial Location, inches
A	8.700
B	9.600
C	9.789
D	10.969
E	11.611
F	13.680
G	14.660
H	15.980
I	16.250
J	16.860
K	17.400
L	17.694

Figure 20 demonstrates the relative convergence which occurred in 60,000 iterations. Note that the predicted exit plane Mach number agrees favorably with the 1D calculation.

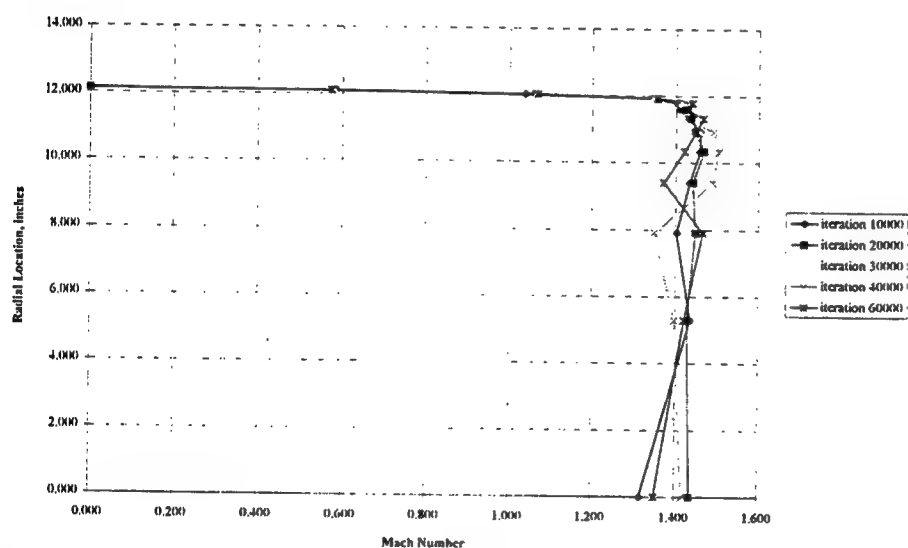


Figure 20. Mach Number Convergence For Test Case #4

Considering Figure 21, it is seen that the cooling wall temperatures also converge within 60,000 iterations. The temperature comparison between experimental values and CFD predictions is illustrated in Figure 22. Here, it can be seen that the predictions were within 8.3% at the location of largest error. The temperatures predicted for the majority of the cooling liner agree with the data to within about 6.0%.

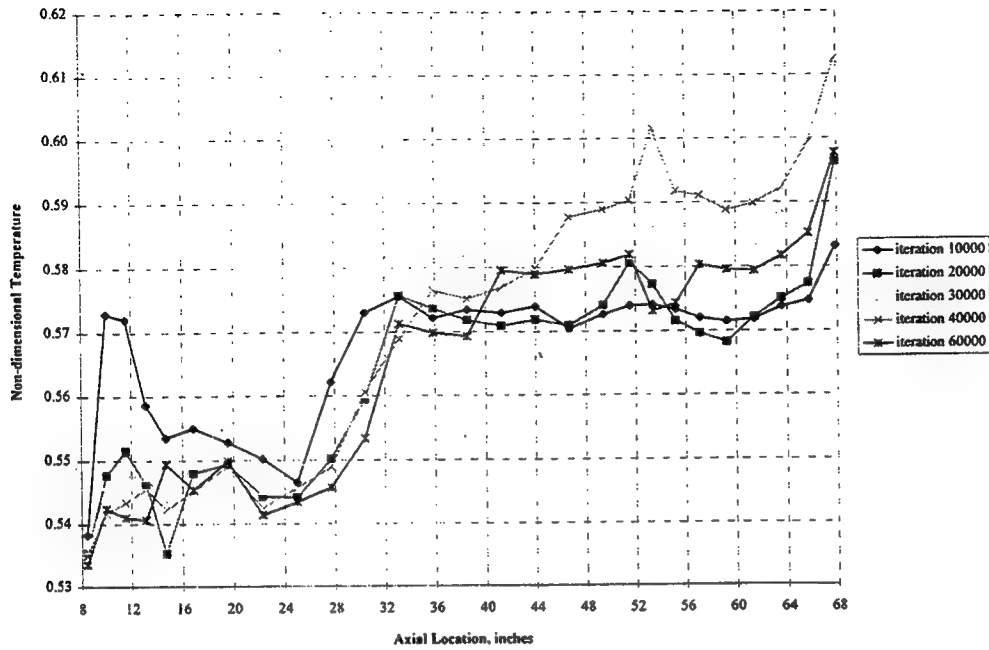


Figure 21. Temperature Convergence For Test Case #4

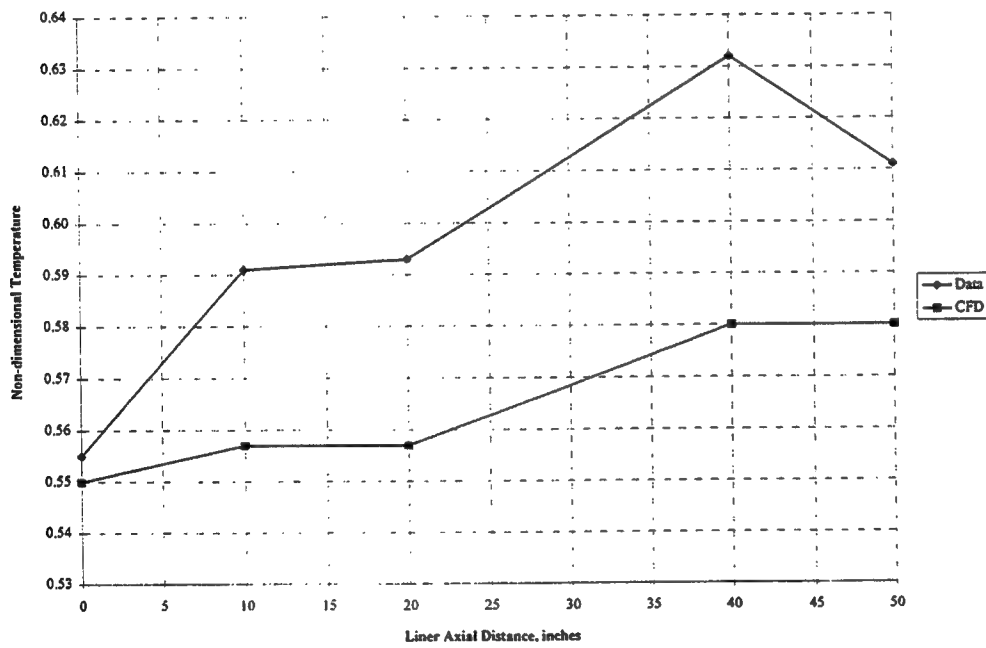


Figure 22. Temperature Comparison For Test Case #4

CONCLUSIONS

The accuracy of representing the flame holders as equivalent 2-dimensional hoops is questionable. It is clear that the level of mixing obtained by using this model is not as significant as the mixing achieved by the actual 3-D flame holders. The CFD tests also established that the placement of the representative blockage areas was crucial in achieving reliable information.

To increase the accuracy of the CFD tests, the type 68 porous, no-slip wall with heat conduction boundary condition needs to be re-evaluated. Initially, the boundary condition was developed from data taken with the porous wall perpendicular to the flow direction. Developing the boundary condition with data taken from an experiment where the wall is parallel to the flow direction may correct the problems encountered during these test cases. If the type 68 b.c. is fixed, it would eliminate the need to use the type 60 no-slip wall b.c. to represent the cooling liner passage.

Finally, the possibility of conducting a full 3-dimensional test case should be investigated. The need for accurate wall temperatures may justify the time required in developing the 3-D model. Through the use of symmetry boundary conditions, the time needed to conduct the test case could be reduced to a level considered to be reasonable.

REFERENCES

1. Ledbetter, G. et al., Predictive Methodology: Approach and Findings for the Prediction of Internal Flow and Surface Temperatures, Internal Memorandum for Sverdrup Technology, AEDC Group, March 1991.
2. Steinbrenner, J. P. and Chawner, J. R., "The GRIDGEN Version 8 Multiple Block Grid Generation Software", MDA Engineering, Inc., Arlington, Texas, December 1992.
3. Sirbaugh, J. R. and Cooper, G. K. et al., "A User's Guide to NPARC", April 1995
4. Walatka, P. P. and Clucas, J. et al., *FAST User Guide*, NASA Ames Research Center: NAS Division, RND Branch, October 1993.

APPENDIX

GRIDGEN²

GRIDGEN Version 9 is a software package consisting of two codes designed to generate structured grids for use in computational sciences. The grids created can be 3-D, single block, multi-block, single surface structured grids, and overset structured grids. The GRIDGEN's two codes are GRIDGEN and GRIDGEN3D. GRIDGEN is an interactive code used to partition a 3-D domain into blocks. It also is used to distribute grid points on surfaces, initialize and refine surface grid points, and to initialize volume grid points. GRIDGEN3D is needed to refine volume grid points.

GRIDGEN allows for the import of a geometry model, such as a NASA-IGES file, or the user can define the appropriate geometry. Once the geometry is established, the user then distributes grid points along the surface components. GRIDGEN supplies various functions to allow for the clustering of grid points anywhere along the surface. After the surface's perimeter is selected, the grids are then initialized. Any grid point refinement required is then accomplished through the use of algebraic and partial differential equation methods. The process for generating volume grids is similar, but the user needs GRIDGEN3D to refine volume grids.

The code has interfaces to various analysis codes including: TEAM, NPARC, OVERFLOW, CNSFV, DTNS, FANS. This allows the user to set solver-specific boundary conditions on the needed surfaces. The grid can be exported in the exact form required by the flow solvers.

GRIDGEN consists of approximately 150,000 lines of FORTRAN and C code, with all graphics implemented in Silicon Graphics' IrisGL graphics language. GRIDGEN runs on Silicon Graphics and IBM RS/6000 workstations configured with the appropriate graphics adapter, a 24-bit frame buffer, and a Z-buffer.

NPARC Navier-Stokes Code³

The NPARC Navier-Stokes code Version 2.1 is general purpose computational fluid dynamics flow solver with a wide range of applications toward aerospace design and analysis. The code is actively supported by the NPARC alliance, a partnership between Arnold Engineering Development Center and the NASA Lewis Research Center.

The program uses specified boundary surfaces and flow conditions to calculate the properties of the fluid flow. The boundary surfaces can be complex and the fluid can be treated generally. Inviscid and viscous flows can be calculated. The viscous flows can be either laminar, turbulent, fully viscous or shear layer type. The NPARC code can be used to simulate both steady-state and transient flows. The program is written in FORTRAN 77 and is easily ported to most computer architectures.

The NPARC simulations have been conducted on varied topics such as supersonic and hypersonic inlet design, rocket nozzle failure analysis, and turbine engine exhaust mixer design. The code has also been applied to other aerodynamic problems including missile cone analysis, ducted flow analysis, and instrumentation probe design.

FAST (Flow Analysis Software Toolkit)⁴

FAST is a software package used to visualize data. It consists of separate modules or programs which allow the user to examine the results from computer simulations. The user can load data files, perform calculations on the data, visualize the results of these calculations, and construct scenes of 3-D graphical objects that may be animated and recorded. The main focus of FAST is computational fluid dynamics, but other types of visualization can be done.

FAST combines the capabilities of PLOT3D, RIP, SURF, and GAS into an environment that shares data. The modules of FAST use a highly interactive user interface. This user environment can be custom configured and new modules can be added. FAST runs only on Silicon Graphics IRIS 4D workstations.

**INSTALLATION AND INVESTIGATION OF ALE3D
AND ITS SUPPORTING PACKAGES**

**Jacqueline C. Haddock
Graduate Research Assistant
Department of Mechanical Engineering**

**The University of Tennessee Space Institute
B. H. Goethert Parkway
Tullahoma, TN 37388-8897**

**Final Report for:
Graduate Student Research Program
Arnold Engineering Development Center
Arnold Air Force Base, TN**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.**

and

Arnold Engineering Development Center

August 1995

INSTALLATION AND INVESTIGATION OF ALE3D AND ITS SUPPORTING PACKAGES

Jacqueline C. Haddock
Graduate Research Assistant
Department of Mechanical Engineering
The University of Tennessee Space Institute

Abstract

Boeing Commercial Airplane Group (BCAG), Lawrence Livermore National Laboratory (LLNL), and Arnold Engineering Development Center (AEDC) have entered a partnership sponsored by JAST to integrate turbomachinery with existing fluid/structure code technology to analyze the structural effects on airframes due to surging engines. AEDC's responsibility is to incorporate existing turbomachinery technology into LLNL's fluid/structure code, ALE3D. The first step in this process was to bring the ALE3D package onsite to AEDC and analyze and evaluate the code system. Upon completion of installation on a Silicon Graphics workstation, the programs were executed for several test cases that were provided to demonstrate their performance for modeling different physical situations, including explosions, implosions, extrusions, fluid flow, and ballistic impacts. Through interaction with LLNL, problems with installation and execution were resolved and many questions regarding the ALE3D algorithm were answered. A demonstration of the plotting package, MeshTV, was presented to key department personnel and other interested parties. Independent meshes were generated using the gridding package, INGRID. A brief oral presentation on all work conducted during the summer program was also given.

INSTALLATION AND INVESTIGATION OF ALE3D AND ITS SUPPORTING PACKAGES

Jacqueline C. Haddock

Introduction

The United States government has initiated a program for combining existing technologies from many disciplines of science and engineering to revolutionize the aircraft industry. Utilizing technology from both private and public industry, the JAST (Joint Air Strike Technology) program hopes to construct a universal aircraft, capable of successfully performing many types of missions, to be employed by all branches of military service. In order for this goal to be accomplished, many problems must be addressed and solutions must be found. One such problem being addressed is the structural effects that surging engines have on airframes. Successful numerical investigation and solution of this problem require consistent coupling of fluid analysis, structural analysis, and turbomachinery simulations. Numerical simulation of turbomachinery and structural interaction may minimize the cost of integrating airframes and propulsion systems by decreasing expensive experimental testing. Additional benefits of such an integrated simulation code would include: analysis of high cycle fatigue engine failures, simulation of complicated engine environments which are difficult to evaluate in test cells, and design phase savings due to the engine and airframe being simulated together.

Therefore, JAST is sponsoring a joint venture between Boeing Commercial Airplane Group (BCAG), Lawrence Livermore National Laboratory (LLNL), and Arnold Engineering Development Center (AEDC) to develop and implement a turbine engine fluid/structure dynamic analysis simulation package. Each member contributes to the project through its expertise in a particular field. As a builder of commercial airplanes, Boeing provides airframe experience and structural test data. LLNL provides technical support through its development of a numerical simulation code, ALE3D, for modeling fluid/structure interactions. AEDC provides knowledge and experience of turbomachinery from extensive testing of commercial and military engines and from its own compressor and engine numerical simulation codes, including the compressor codes DYNTECC and TEACC and the engine codes ATEC and ATEST.

AEDC's responsibility is to incorporate turbomachinery into the existing fluid/structure interaction code, ALE3D, developed by LLNL. This report focuses on the initial stage of this project at AEDC.

Approach

First, ALE3D and its supporting packages were received from LLNL and installed on a Silicon Graphics Iris (SGI) workstation. A script for building the system was not available; consequently, the installation was largely "hands on". A specific file structure had to be maintained in order for the executable to build correctly. Several libraries and pre-processors had to be built as well. Following the release notes that accompanied the code was challenging, but with perseverance and the aid of Mr. Scott Futral of LLNL, the installation process was correctly completed in a few days. The plotting package, MeshTV, and the mesh generator, INGRID, were somewhat easier to install because, unlike ALE3D, there were no separate libraries or pre-processors to build. As a result, ALE3D, MeshTV, and INGRID are currently available on a SGI workstation onsite at AEDC. Additional observations and explanations were recorded during the installation process to hopefully circumvent any difficulties or confusion arising during future installations of this software on workstations at this site.

Next, ALE3D was executed for the nineteen test cases provided by LLNL. The execution time required was influenced by the size and complexity of the mesh as well as the physics of the problem and ranged from less than a minute to over eight hours. The results produced by the code were viewed using MeshTV; visualization and manipulation of the test case data were achieved with relative ease through MeshTV's graphical user interface. In this manner, some of ALE3D's capabilities were demonstrated while exploring the plotting capabilities of MeshTV. This type of evaluation technique was applied to all of the test cases. A brief demonstration of MeshTV was presented to key departmental personnel and other interested parties.

Finally, research into ALE3D provided answers to some questions about the algorithm, and mesh generation was demonstrated using the coordinate generator, INGRID. Normally, the grid generation would precede the use of ALE3D and MeshTV, but grids for the test cases were already made. Therefore, the use of INGRID was not necessary at the beginning of my research. An effort to generate experimental

meshes using INGRID met with some difficulties due to the complexity of the code, but grid restrictions were successfully identified.

Description of the Simulation Package

The ALE3D code system consists of a coordinate generator, a restart file generator, the main physics code, and post processors such as MeshTV. The coordinate generator, currently INGRID (also a product of LLNL), generates the mesh, assigns material numbers to elements, and designates boundary conditions. The restart file generator assigns material properties, selects high explosive burn models, creates material boundaries that do not conform to element boundaries, and sets various execution options for the main code; it reads the INGRID output file and a separate generator input file containing equations of state, constitutive models, etc. The main code is executed by specifying a restart file name and an ASCII command input file (CIF) name. The CIF directs the main code to generate plot files, restart files, and time history files. The code can also be interactively queried from the keyboard during execution. MeshTV is the plotting package provided with code; it is used to view and analyze the results produced by ALE3D. The utilities ULTRA II, PDBView, and PDBDiff are also included with the simulation package. These utilities are used to view the time history files, edit restart files, and compare files, respectively.

Features of INGRID

INGRID is a generalized three-dimensional finite element mesh generator for modeling non-linear systems. It provides the user with the ability to generate large, complex geometrical models using beam, shell, and eight-node hexahedral brick elements. The wood screw shown in Figure 1 is an example of a fairly complex mesh generated using INGRID. Figure 2 shows the preferred types of elements used

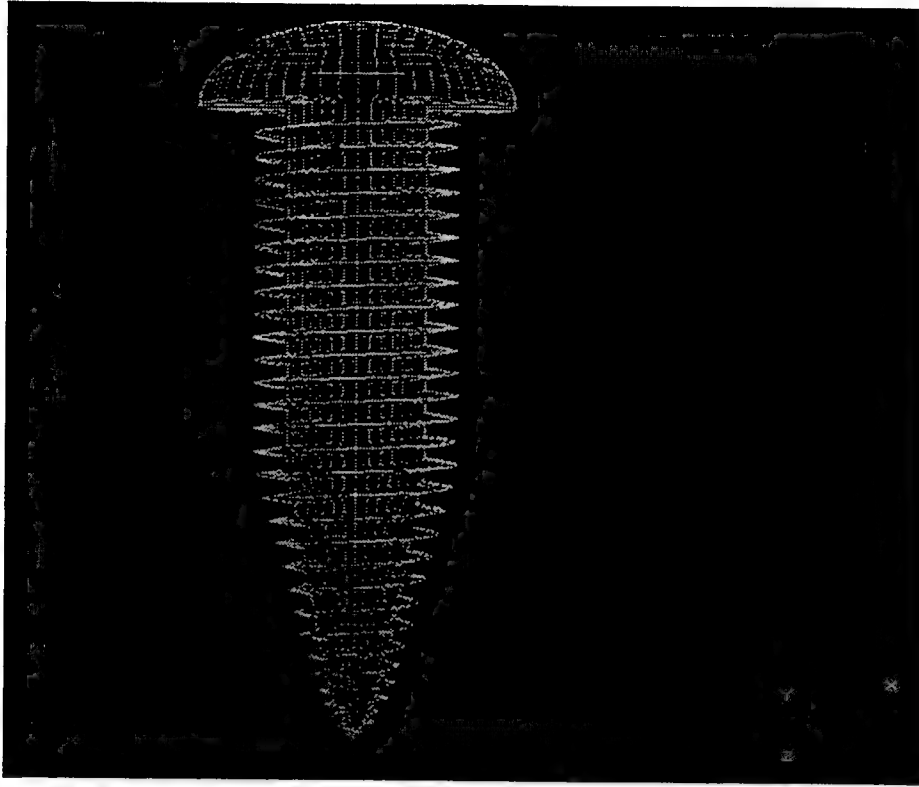


Figure 1 A wood screw

for building meshes. Other elements such as tetrahedra and degenerate (wedge-shaped) elements are possible but should be avoided due to their tendencies to produce inaccuracies in calculations. Slide surfaces are used for boundaries through which disjoint parts of the grid interact. ALE3D uses INGRID to assign coordinates and velocities to nodes and material numbers to elements and to designate boundary conditions. It also generates the connectivity matrix correlating nodes and elements.

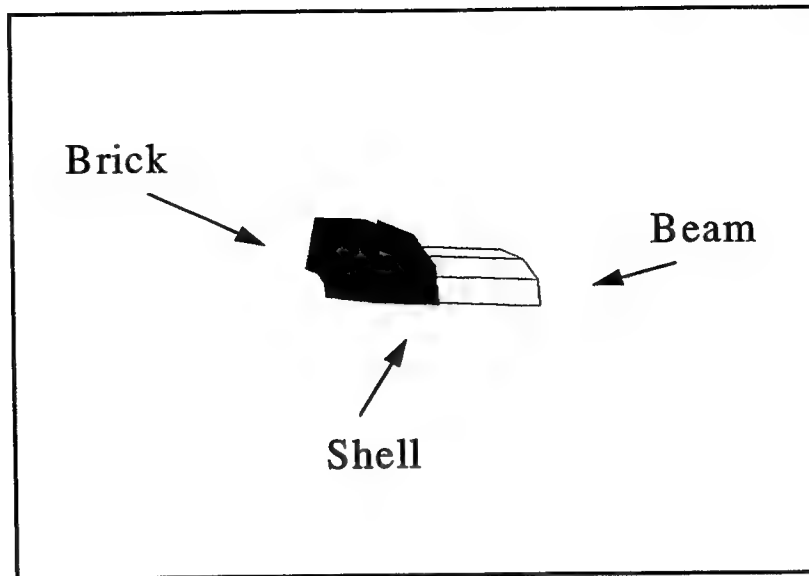


Figure 2 Examples of brick, shell, and beam elements

Features of ALE3D

ALE3D is an explicit, three-dimensional arbitrary Lagrangian-Eulerian hydrodynamics code with material strength. As previously mentioned, the primary interest in this particular code is its ability to model situations involving fluid-structures interactions; however, ALE3D also models situations involving low stress metal forming, high temperature high explosives, and objects undergoing gross deformations due to shocks. The nature of the arbitrary Lagrangian-Eulerian algorithm provides the capacity to operate in Lagrangian mode, Eulerian mode, or an intermediate mode. In the Lagrangian formulation, coordinates are fixed with the material and convect with it, but in the Eulerian formulation, the coordinates are fixed in space. Hence, Lagrangian formulations are traditionally well suited for solid mechanics while Eulerian formulations are traditionally used for fluids. However, in some situations the reverse might be more appropriate, with Lagrangian coordinates applied to fluids and Eulerian coordinates used for solids. Therefore, the ability to use Lagrangian, Eulerian, or mixed Lagrangian-Eulerian greatly expands the possibilities for computation. For instance, studies involving both fluids and structures might use a mixed Lagrangian-Eulerian mode.

The code is executed by initially performing a complete Lagrangian computation succeeded by an advection step. For each calculation cycle, nodes are accelerated and the velocity and position at the end of the cycle are calculated. This motion distorts the elements defined by the nodes, and the resulting changes in volume, work due to the volume changes, and new pressures are calculated. This is the Lagrange step. After completion of the Lagrange step, a new representation of the mesh is created and fluid variables are reevaluated on the new mesh by allowing the fluid to flow through the grid; this is the advection step. The mode that is being used determines whether one or both of these steps is necessary. Stress gradients and strain rates for the Lagrange step are evaluated using a finite element method, and a monotonic advection algorithm is used in the advection step. The new mesh positions are calculated using either the finite element-based equipotential grid relaxation method or the Eulerian grid relaxation method. If the problem is run in the Eulerian mode, the Eulerian method is automatically used.

The main code produces a sequence of plot files, optional time history files, and an ASCII file containing a cycle-by-cycle edit of the problem. The plot files are PDB (Portable Database) files with imbedded information describing the meshes and mesh variables and are currently the only type of files accepted by the post-processor, MeshTV. The time history files may be analyzed with the ULTRA utility included with ALE3D.

Features of MeshTV

MeshTV is an interactive program for graphing and analyzing scientific data. In particular, it is used to view output from ALE3D; however, any application that chooses to write its output in the appropriate format can use MeshTV for post-processing. MeshTV operates on many different types of meshes, including one-, two-, and three-dimensional quadrilateral and unstructured meshes, and produces numerous types of plots, including material boundary, contour, vector, mesh, projected surface, and value versus distance plots. Furthermore, there are many data operators and plotting options to control the appearance and production of plots. Up to sixteen independent graphics windows are supported at all times. Two of MeshTV's more unique features are animation and support of mixed material elements.

MeshTV is composed of two parts: 1) the "heart" program which is responsible for performing all the graphics and analysis work and 2) the "standalone application" which provides the graphical user

interface (GUI) for the "heart" and is typically the way MeshTV is invoked. A command line interface and a command input file may also be used to communicate with MeshTV. The command input file is particularly useful for setting up animations.

Other Utilities

Other utilities included in the ALE3D package are: ULTRA II, PDBView, and PDBDiff. ULTRA II is a program for the presentation, manipulation, and analysis of one-dimensional data sets. Specifically, this application is used to process time-history files produced by ALE3D. PDBView is a program for interactively browsing PDB files, displaying the contents graphically or textually. PDBView can be used to examine or modify any variable in the restart file, but the primary use is to remove elements and slide surfaces from ALE3D calculations. PDBDiff is a program for comparing the contents of two PDB files and displaying their differences.

Capabilities

The capabilities of each application is best viewed through graphical visualization. The focus of the research was on the capabilities of MeshTV and to use it as a window for viewing the results of ALE3D. No attempt was made to justify or validate the results of ALE3D.

Figure 3 shows a two-dimensional orthogonal slice of the flow of a fluid around a plate. The fluid enters through one boundary and flows out the opposite boundary. The shading represents the speed of the fluid. This case demonstrates the use of a vector plot in MeshTV. A two-dimensional slice was required because MeshTV's vector plotting capability is limited to two dimensions. LLNL used this problem as a test of inflow and outflow boundary conditions.

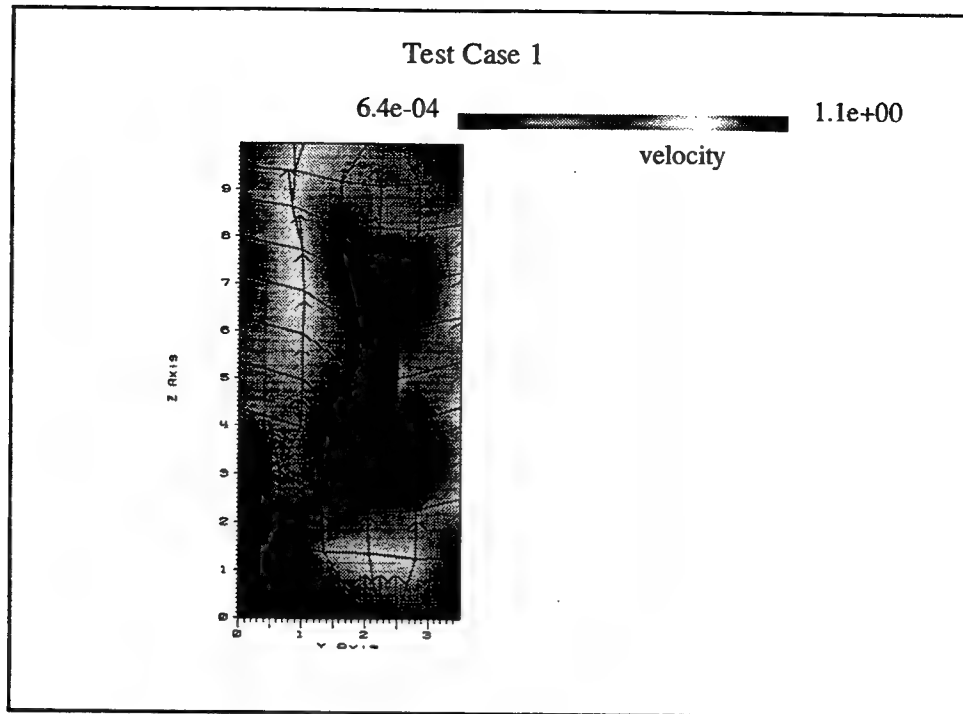


Figure 3 Flow over a plate

An extrusion process is shown in Figure 4. The working fluid, aluminum, is pushed through a die into a low density gas (void). The shading represents the density of the different materials. This geometry is more complex than the flow around the plate. Slide surfaces separate the working fluid and void from the die. This problem was used by LLNL to test advection and intersecting slide surface logic in a fairly complex geometry.

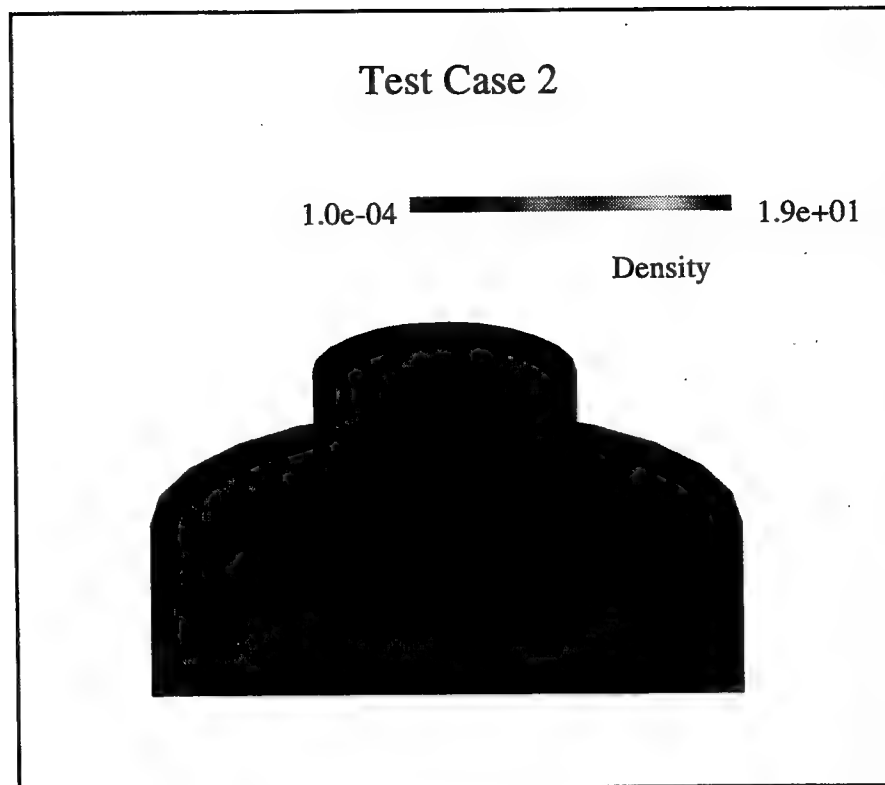


Figure 4 An extrusion process.

Future Work

The installation and investigation of ALE3D and its supporting packages was an overall success. Several questions regarding the forms of the basic equations and solution technique used by the algorithm remain unanswered at this time. However, a request has been made to LLNL seeking answers to these questions. Through JAST funding, work with this simulation package will continue at AEDC. With the help of BCAG and LLNL, an engine/airframe configuration will be identified, and an appropriate mesh will be constructed using INGRID. Then, turbomachinery will be incorporated into the ALE3D code, and calculations will be performed. Finally, the results produced will be compared to test data to evaluate the effectiveness of the algorithm.

References

Anderson, Steve, et al., *Users Manual for ALE3D (An Arbitrary Lagrange/Eulerian 3D Code System)*,
Lawrence Livermore National Laboratory, Livermore, CA, 1995.

Christon, Mark A. and Donald Dovey, *INGRID: A 3-D Mesh Generator for Modeling Nonlinear Systems*,
Lawrence Livermore National Laboratory, Livermore, CA, UCRL-MA-109790, 1992.

ENVIRONMENTAL PLANNING METHODOLOGY

Jan S. Lach
Graduate Student
Department of Engineering Management

University of Tennessee Space Institute
B. H. Goethert Parkway
Tullahoma, TN 37388-8897

Final Report for:
Summer Graduate Student Research Program
Arnold Engineering and Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Arnold Engineering and Development Center

August 1995

ENVIRONMENTAL PLANNING METHODOLOGY

Jan S. Lach
Graduate Student
Department of Engineering Management

Abstract

The purpose of this research was to develop a methodology for building an environmental plan that can be used by management to identify, select, and prioritize environmental projects. The research procedure was to 1) examine the methods recommended by management analysts for developing plans, 2) examine the environmental planning experiences of government and industry, 3) examine this process as it is practiced on an installation, and 4) combine this information into a practical methodology that would provide a coherent, proactive plan for achieving the best value from environmental compliance projects. A functional plan that is structured to be consistent with the "pillars" or categories of Air Force environmental management and with the existing management information system is developed. The methodology uses a structured decision making process to identify environmental improvement projects and is developed as a five step procedure: 1) develop a statement of need; 2) appraise project alternatives; 3) perform a weighted strength assessment; 4) prioritize projects within their pillar; and 5) prioritize projects overall.

ENVIRONMENTAL PLANNING METHODOLOGY

Jan S. Lach

1.0. Introduction.

In recent years, environmental management has received a great deal of attention within the Air Force. Due to the industrial activities performed at most bases and the aging facilities that must be used, achieving and maintaining environmental compliance has become a significant effort. The key to successful environmental management is planning, both to ensure environmental goals are met and to provide a coherent, cost effective method for achieving those goals.

The environmental management division at Arnold Engineering and Development Center (AEDC) located at Arnold Air Force Base in Tennessee represented an excellent opportunity to do this research. AEDC is a large industrial installation with environmental responsibilities that range from forestry and wildlife management to industrial waste treatment and they were in the process of developing a plan to manage environmental programs.

2.0. Planning

2.1. The General Business Planning Process. The generally accepted business planning process consists of five steps: establishing the business vision and mission, setting goals and objectives, developing the plan strategy, implementing and executing the plan elements, and evaluating the plan performance and making adjustments. Establishing the business vision and mission is fundamental to plan development and execution. These are statements of the purpose, direction, and intent of the business and by extension, the plan.

Translating the business vision and mission into measurable and attainable goals and objectives is crucial to plan execution and evaluation. In this paper, a goal is defined as an end result, while an objective is the means to that end. An objective may consist of a single performance target or a group of performance targets that contribute to achieving a goal.

The plan strategy defines the purpose and scope of the plan, why the plan is important, how the plan will be implemented, and the roles and responsibilities of those who will execute the plan. The existing structure of

the organizations that will be involved must be incorporated into the plan.

Implementing and executing the plan requires building an organization that can carry out the plan strategy, creating a corporate culture that is conducive to successful plan implementation, motivating people to want to achieve the business goals and objectives, and developing internal systems that support the plan strategy¹.

Evaluating the plan performance is a necessary part of a continuous improvement process. Once established, a plan should be a living document that is continually adjusted to reflect the changing business environment.

2.2. Business Plan Structure. The overall plan development for a business can be visualized as a layered pyramid of plans each with an increasingly broad application (exhibit 1)². Each level of the pyramid represents not only the planning level but also the organizational level responsible for developing and executing the plan. For the purposes of this research, the business has been modeled after a single-business company, the business of interest being environmental management. When taken together, all three plans create the environmental business plan; success of the overall plan is dependent not only on the success of each plan individually, but on the successful integration of the plans as a whole.

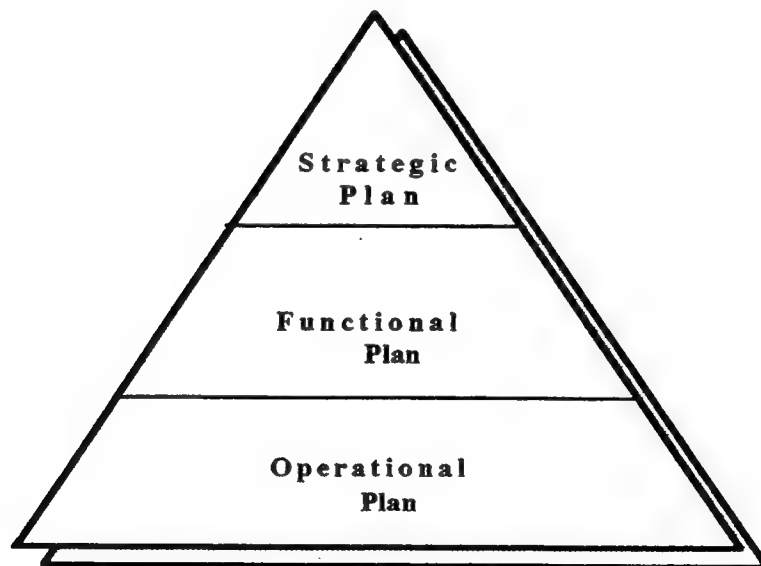


Exhibit 1. The business plan pyramid (adapted from Thompson and Strickland).

The strategic plan forms the apex of the pyramid. This plan establishes the environmental vision and mission of upper level management (executive and directorate). In the case of an Air Force base, the strategic goals and objectives will be derived from Department of Defense and command guidelines; federal, state, and local regulatory requirements; and installation policies. This plan has a long term focus and a planning horizon of several years. Successful execution of the strategic plan should infuse the installation with a sense of mission and purpose and demonstrate strong leadership.

The functional plan forms the middle section of the pyramid. This plan establishes how the strategic plan will be implemented by each functional department or division, and has a short term planning horizon of one to three years. The functional plan will be discussed at length in this paper.

The operational plan forms the base of the pyramid and addresses the day-to-day activities that must occur to achieve the goals and objectives of the strategic and functional plans. While the functional plan is primarily concerned with identifying and implementing the physical improvements needed for proper environmental management, the operational plan is concerned with guiding the activities of installation personnel to maintain environmental compliance.

2.3. SWOT Analysis. A first step in generating a successful business plan is to analyze the strengths, weaknesses, opportunities, and threats that form the business environment of the organization³. This is the familiar SWOT analysis used extensively in business. In environmental planning, strengths are areas where compliance is good and there are no imminent violations. Weaknesses are areas where there is a lack of information about environmental conditions. Opportunities are areas where timely intervention could prevent future problems. Threats are areas where there are current or imminent violations.

Performing this assessment enables the plan makers to describe the existing situation and the areas that need attention. SWOT analysis requires information concerning current and forthcoming regulatory and policy requirements, existing plant and equipment, current processes, and the current status of improvement efforts. The SWOT analysis is a continuous process, subject to the inflow of information, the constantly changing requirements of environmental compliance, and the revised conditions that occur as the plans are implemented and projects are completed.

3.0. Failures in Planning.

3.1. Failure to Consider Execution. The most common reason for failure of plans in general is failure to let those who will be responsible for execution do the planning⁴. This creates a situation in which no one can be held accountable for the success or failure of the plan. Only those people who have to actually do the work can plan how to do the work; allowing them to do the planning gives them ownership of the plan and will motivate them to ensuring success.

3.2. Shortfalls in Environmental Planning. Harkness, Marchlik, and Polito⁵ identify four key shortfalls that occur in both public and private sector environmental planning: failure to establish a clear environmental vision and provide effective environmental leadership; not taking the organizational structure of the installation into account; not considering environmental issues early in projects and planning for them; and having an insufficient information management system.

3.3. Failure to Establish a Clear Environmental Vision. It is the task of upper management to establish the environmental vision and mission for the installation and to lead the effort to achieve environmental goals. A good measure of management's success in this regard is that each employee should know exactly how he or she fits into the plan and how what they do affects not only their own success, but also the success of the installation.

3.4. Failure to Account for Organizational Structure. Organizational structure is often overlooked when making plans⁶. The manner in which information flows into, through, and out of an organization; who influences the decision making processes; who does the work; and how the work is done directly affects the success of planning in that organization. There must be integration and compatibility between the planning system, the organizational structure, and the strategic situation of the organization⁷.

In Air Force Materiel Command (AFMC), there are two environmental management structures used both at headquarters and the base level. At the Air Logistics Centers, environmental management is a directorate under the center commander (the EM structure). At AEDC, Headquarters (HQ) AFMC, and most of the small bases, environmental management is under the civil engineering directorate (the CEV structure). Andreoli⁸ argues that how the environmental management function is placed in the overall base

organizational structure directly affects the relationship between the base and outside regulators, between functional areas on base, and the ability of the base to achieve and maintain environmental leadership. Andreoli states that most problems in base environmental management are a result of lack of access to upper management and lack of positional authority (the environmental function is not positionally equal to operations and maintenance at the directorate level) and that these problems can be effectively eliminated by adopting the EM structure.

3.5. Failure to Ensure All Environmental Issues are Addressed. A system for ensuring that environmental issues are addressed properly in all projects must be included in the plan execution. The projects at risk are those that are not ostensibly environmental projects, such as general facility improvements, routine equipment substitutions, or changes in industrial processes or materials. The practice at most installations that were reviewed for this paper is to rely on whoever is in charge of non-environmental projects to identify those situations where environmental concerns may become an issue. This creates a problem in the current climate of rapidly changing environmental policies because those who are not involved with environmental issues on a daily basis may not be aware of all the requirements. A few installations require the environmental engineering division to review all specifications and drawings and include a signature block for the environmental department to sign on all drawings.

3.6. Failure to Employ a Sufficient Management Information System. Lack of an adequate information management system can doom any plan to failure from the start. There must be a constant flow of pertinent information into and through the organization to keep everyone aware of what is going on. The Work Information Management System (WIMS) used by the Air Force is already a substantial part of the information that flows from the base to headquarters and to the Air Force but there must also be information that flows from the various installation facilities to base environmental management. A procedure that has been implemented by the Idaho National Laboratory to gather information from individual facilities is the process waste assessment (PWA)⁹.

The basic process used in performing a PWA is to draw a control volume around the physical boundaries of a facility, identify and measure everything that goes in (materials, chemicals), identify what happens to it in the facility,

and identify and measure what comes out (stack gases, combustion products, effluent, materials, chemicals). The resulting waste stream characterization can then be used to identify source, composition and type, disposal requirements, rate and quantity generated, percentage of total installation waste generated, and all the various associated costs. An important aspect of the PWA is that the facilities must be analyzed with regard to everything that occurs within the facility, whether directly or indirectly related to the facility mission. For example, cleaning, painting, and other general maintenance work must be added to the PWA if it affects the facility waste stream.

The information acquired through the PWAs is crucially important because it is the only way to know what really happens on the installation from an environmental standpoint. For example, the process at a certain test facility may cause ethylene glycol to show up in the waste water stream from that facility. By knowing how much ethylene glycol went into the facility and what processes used it, it is straightforward to determine how much ethylene glycol was lost and where it most likely came from. This not only makes troubleshooting and repair much easier and less costly, it also creates the background for analyzing the process with a view to material and/or process substitutions that minimize use and waste. The PWA can also be used as a basis for establishing and monitoring process metrics.

PWAs are also valuable in identifying those facilities that contribute most to the waste stream, whether in air, water, or soil. These are the facilities where the most cost effective use of environmental funds will be achieved. The PWA process should begin with the most environmentally critical facilities and continue until all installation facilities have been assessed.

An example of a PWA-type program is the HAZMAT program originally developed by the Navy at Point Mugu and now adopted by the Air Force¹⁰. At AEDC, the HAZMAT Pharmacy and solvent replacement elements have been combined with a Just-In-Time supply policy to track and control a hazardous substance from the time it arrives on base to the time it leaves and is disposed of. The Navy found the program at Point Mugu so successful that it has been expanded to include petroleum products and the installation recycling program.

4.0. The Functional Plan.

4.1. Purpose. The purpose of the functional plan is to implement the strategic plan. The first task in creating the functional plan is to define the external and internal business situation that influences base environmental management. External conditions that influence base environmental management include federal, state, and local regulations, and Air Force and command directives. Internal conditions that influence base environmental management are base policies, the organization's customers, and the organizations, and base operations. The groups that will be affected by and participate in the plan must be clearly identified.

4.2. Limiting Conditions. For Air Force bases, the primary external limiting conditions are the project funding structure and project priorities used by the Air Force and the annual funding levels which are determined by HQ AFMC. WIMS is used to track real property projects on all bases. Environmental management has a separate section in WIMS that tracks only environmental projects and funding, called the A-106. The A-106 is broken into four categories which are known as the "pillars" of environmental management¹¹: environmental compliance, installation restoration, pollution prevention, and resource conservation. Each "pillar" has its own prioritization method¹². In general, each category has a priority for violations, a category for imminent violations, and a category for projects that are above what is required ("nice to haves"). Projects are first identified, categorized, and prioritized at the base level, then validated by the Air Force.

Once validated, project funding is determined by headquarters. The current policy is to fund projects that eliminate violations and to fund as many imminent violation projects as the budget allows. Projects that increase compliance above what is required are not funded; thus effectively eliminating pollution prevention projects from funding consideration¹³. Since the WIMS A-106 system is the primary funding vehicle for environmental projects, the restrictions of this system must be incorporated into the functional plan. It is not cost effective to spend a lot of time identifying and scoping projects that have little chance of funding. While these projects cannot be ignored entirely since they may develop into violation elimination projects in the future, any efforts beyond an initial scope and rough order of magnitude cost estimate are not warranted.

4.3. **Prioritizing Projects.** The bulk of planning activities at the functional level are concentrated on identifying, selecting, and prioritizing real property improvement and equipment replacement projects. Since projects with outstanding or imminent violations have funding priority, it is in the best interest of the base to achieve the greatest number of environmental objectives through these projects. Environmental projects should be identified, scoped, and prioritized within each pillar several times a year, to ensure that the list is kept current. Performing this process continuously will also make it possible to quickly identify worthy projects if additional funding suddenly becomes available, thus relieving the scramble to identify projects for fallout funding at the end of the fiscal year, which is typical of most bases.

Once prioritized within their respective pillar, the projects would "compete" between pillars for inclusion in the annual funding request. Since projects that address current violations are usually funded, the competition for funding is mostly between projects that address imminent violations. For example, the top two imminent violation projects from each pillar would be prioritized into a list of eight projects that would be submitted for funding. If there is adequate funding for only seven projects, it is straightforward to select which project must be eliminated. This project would then be the first choice for end of fiscal year funding or, failing that, would compete again the following year.

5.0. Functional Plan Methodology.

5.1 **Structured Decision Making.** Structured decision making starts with identifying what Blanchard and Fabrycky¹⁴ term the limiting factors and strategic factors that influence the decision. Limiting factors restrict accomplishment of the objective, while strategic factors are those limiting factors that can be altered to make progress possible. The funding limitation imposed by the A-106 system is the most obvious limiting factor, but this limitation can be transformed into a strategic factor by carefully scoping projects. Once identified, these factors are used to define the boundaries of the decision making activity, i. e., what cannot be done, what can be done, and what can be changed.

A structured decision making process should be used to screen, select, and prioritize alternatives and projects. Souder¹⁵ has identified several common roadblocks that can work against any decision making process. These

roadblocks are: jumping from problem to solution without analysis, premature evaluation before the problem is completely identified (in which case the wrong problem may be solved), over-reliance on experience, premature commitment (the quick fix solution), confusion of problems and symptoms, focusing on one solution when several solutions may be required, and overlooking how the decision will be implemented. These roadblocks and their obvious results can be largely avoided by developing and using a structured decision making process. The process involves several sequential steps as demonstrated in the flowchart shown as Exhibit 2.

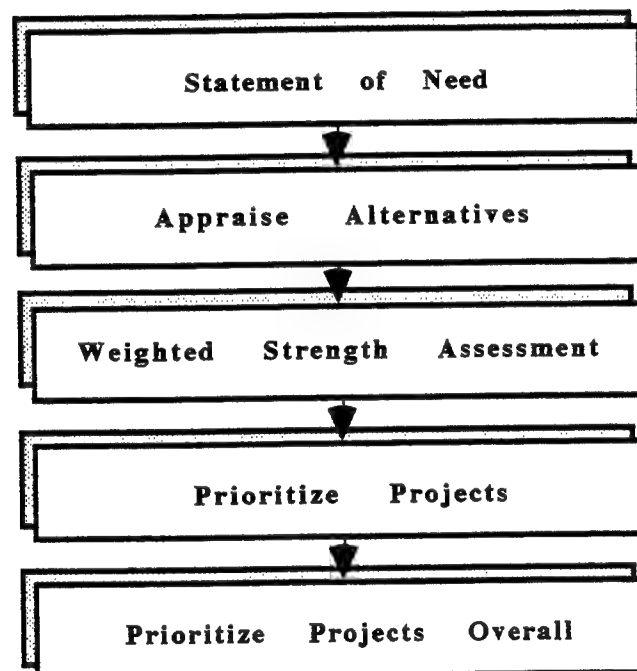


Exhibit 2. Project selection and prioritization methodology.

5.2. The Statement of Need. Project identification begins by developing a statement of need. The statement of need may be submitted by anyone on base and should explain in broad terms the project objective, how the project objective fits in with the strategic and operational plans, and when the project must be completed. The necessity for a project can be determined either reactively or proactively. A reactive approach to project identification involves identifying discrepancies between the current situation and the required situation or identifying deficiencies between present and past

performance. Proactive project identification involves a deliberate search for opportunities by measuring the installation's current performance against other similar installations (benchmarking)¹⁶ or identifying new and emerging technologies that will have the most impact on the installation in the near future. An installation similar to AEDC that was reviewed for this research is the NASA Lewis Research Center.

5.3. Appraisal of Alternatives. Once the statement of need has been issued, alternatives can be identified and appraised. There may be several ways to achieve the project objective, for example, a certain requirement may be met by either changing an industrial process, building a new facility, or modifying an existing one. Each alternative must be scoped and a rough cost estimate performed and then evaluated on an equivalent engineering economy basis, assessed with regard to risk and sensitivity and the best alternative chosen. Once the best alternative has been chosen for each project, a detailed cost estimate must be performed for use in prioritizing the projects.

Economic equivalence can be established using several methods, such as present worth, annual cost, payback period, return on investment, or benefit/cost ratio. Each method has advantages and disadvantages and using a single method alone can yield misleading results. Methods which require assessment of benefits (such as payback period, return on investment, and benefit/cost ratio) must be used with care in circumstances when it is difficult to measure tangible benefits and therefore subjective financial measures of intangible benefits must be used.

Tangible benefits are such things as profits, cost reductions, and fines avoided. Intangible benefits include good will, quality of life, safety, and environmental impact. The procedure for measuring intangible benefits is to place a value on the benefit by assessing how much one is willing pay to acquire that benefit. This type of subjective assessment is liable to vary widely due to differing points of view and can be difficult to defend.

Present worth and annual cost methods do not require assessment of benefits, but do require determination of the investment interest rate (accounting for inflation) for the life of the project. This is usually done either by estimating the interest rate that would be received if the money were invested or by arbitrarily determining the minimum acceptable rate of return.

The decision making conditions that affect each alternative whether certainty, risk, or uncertainty must be included in the equivalent economic analysis and assessed using techniques specific to each condition. Decisions made under certainty require that each alternative has one and only one outcome and election of project alternatives is straightforward to evaluate. Alternatives can be ranked according to whatever criteria is desired, such as lowest annual cost or highest rate of return, and the best alternative chosen.

Decisions made under risk require that the alternatives have more than one outcome and that the probabilities of the outcomes are known or can be determined. Risk can be measured either by determining probabilities objectively or subjectively¹⁷. Objective probabilities are the proportion of times an event has occurred out of the total times it could have occurred. Subjective probabilities are determined by soliciting "expert" opinion as to the chances of a particular event occurring. Subjective probabilities can be determined by conference among experts, by management decision, or by using the Delphi technique¹⁸ in which "experts" are asked to respond to a series of gradually more specific questionnaires about their opinions concerning a particular topic.

A widely used technique for evaluating decisions made under risk is the decision tree. A decision tree is a graphic-analytic approach that enables analyst to see what decisions must be made, in what sequence they must be made, and how they are all interrelated¹⁹. Decision trees are particularly valuable in identifying when more information is needed and how much should be paid for that information.

Decisions made under uncertainty occur when the alternatives have more than one outcome, but the probabilities of these outcomes are either unknown or cannot be determined. Decisions made under conditions of uncertainty require more abstract analytical techniques²⁰.

It must be emphasized that none of these decision making techniques are guaranteed to make the best choice. The alternatives can be evaluated and ranked using qualitative and quantitative techniques, but the final decision is still the responsibility of the person making the decision.

An important point that must be considered when choosing between alternatives is sensitivity to change. Sensitivity analysis is conducted by measuring the change that occurs when a variable (or several variables) of interest is increased by one unit. For example, the interest rate or the

planning horizon could be increased or decreased and the resulting changes in the decision analysis noted. If the outcome is highly sensitive to slight changes in a variable and the choice of alternative is dependent on that variable, then a decision must be made as to the acceptability of this situation. Highly sensitive alternatives should either be re-scoped or eliminated.

5.4. Weighted Strength Assessment. Mathematical decision making is useful when alternatives can be evaluated using strictly economic criteria; however, it is not often the case that environmental projects can be selected purely on financial grounds. A decision making tool that combines both quantitative economic criteria and qualitative analyses of intangible benefits is the weighted strength assessment²¹ (Exhibit 3), in which a set of criteria or key factors is identified with each criteria assigned a weight based on its value relative to the other criteria. Each alternative is then ranked on an appropriate scale according to how well it meets the criteria. The score for each alternative is the total sum of the products of each weight and rank for that alternative. The alternatives are then ranked overall according to the total score.

<u>Key Factor</u>	<u>Weight</u>	<u>Alternatives</u>		
		<u>A₁</u>	<u>A₂</u>	<u>A₃</u>
Initial Cost	10%	3	2	1
Annual Cost	20%	2	3	3
Command Goals	20%	2	1	2
Base Goals	20%	2	3	2
Mission Impact	20%	3	3	3
Risk/Sensitivity	<u>10%</u>	<u>1</u>	<u>2</u>	<u>1</u>
Sum of Weights	100%			
Rating		2.2	2.4	2.2

Exhibit 3. Weighted strength assessment. Rating scale for cost and risk/sensitivity and mission impact: 1 = high, 3 = low; rating scale for goals: 1 = low, 3 = high; Score for each alternative = Σ (weight)(rating). A₂ would be chosen.

5.5. Project Prioritization. Projects can be prioritized within their respective pillars (once the best alternative for each project has been determined) using the weighted strength assessment technique. After the projects have been prioritized within the pillars, projects from each pillar must be prioritized overall for inclusion in the annual budget request (the A-106).

The annual project list can be prioritized by beginning with all the first priority projects from each pillar. This will yield four projects with priorities of one through four. Repeat the process with the second priority projects which will yield projects with priority five through eight and continue until all projects have been prioritized.

5.6. Summary of Methodology. The process of analyzing each proposed project based on the statement of need, the alternative appraisal procedure, the weighted strength assessment, and overall project prioritization yields a functional plan. This analysis should be performed by a team with representatives from the four pillars. The prioritization of the projects within the pillars should be performed by the team responsible for that pillar. The structured decision making techniques used to select the best alternative for each project should also be used to select and prioritize projects.

6.0 Conclusions.

Environmental planning at the base level can be modeled as a business plan for a single business company. The environmental business plan should consist of three different plans: the strategic plan, the functional plan, and the operational plan. The functional plan for base environmental management should focus on identification, selection, and prioritization of projects that either resolve or prevent compliance violations.

Process waste assessments should be used as vehicles for providing the information needed to identify projects and to improve processes to reduce pollution. PWAs can also be used to identify the facilities that contribute most to the waste stream; environmental improvement projects at these facilities will achieve the most for the money.

The functional plan should be divided into four categories: pollution prevention, resource conservation, environmental compliance, and installation restoration to align the plan with the WIMS A-106 requirements

for projects. This will allow the planning efforts to be directly related to the primary project funding instrument.

A structured decision making approach should be used to select between project alternatives, and to prioritize projects. This should not only increase the efficiency of the project selection process, but also serve to fit each project into the business plan of the base.

The four pillars structure can be used to increase the efficiency of the project selection and prioritization process and to maximize the knowledge of the personnel within each pillar by channeling the flow of information into and out of the pillars. This could be accomplished by organizing the personnel in each pillar as a team. This approach would take advantage of the knowledge and experience of personnel and ensure communication between pillars, thus preventing "stovepiping", and reducing the possibility that a solution in one pillar causes a problem in another.

Although the statements of need could be submitted by any organization on base, the identification of project alternatives, and selection of projects would be performed by the pillar team leaders as a group. Project alternatives can be identified and explored from four different points of view, for example, an identified need may be resolved by using either an environmental compliance project or a pollution prevention project. The team leaders would select the best project alternative using the weighted strength assessment. The work of project scope, cost estimates, and prioritization within the pillar would fall to the pillar team. Annual overall project prioritization would also be performed by the pillar team leaders.

The organizational structure of the base and of the environmental management business unit must be incorporated into the business plan. The organizational structure of the base will have a profound effect on successful project execution. Each base must assess whether environmental management is of equal or less importance than operations and maintenance and organize accordingly.

7.0. Recommendations.

1. Develop the three components of the overall plan in order of their scope, i.e. develop the environmental management strategy plan first, the functional plan second, and the operational plan third.

2. Each plan should be developed as a project in its own right. A project manager should be assigned, and a budget, schedule, and milestones should be developed and adhered to. It is recommended that the project manager not be a functional area manager, rather the project should be sponsored by the functional manager because the plan development project will require a concentrated commitment in time and effort.

3. Staff the plan development team with those who will be responsible for implementing and executing the plan. This will establish ownership of the plan and motivate personnel to make the plan work.

4. Perform the process waste assessments concurrently with plan development, incorporating information as it is received.

5. Structure the functional plan to comply with the A-106 WIMS structure and use the violation classifications to aid in project prioritization.

6. Implement a structured decision making process to choose and prioritize projects. The weighted strength assessment appears to be the best choice for environmental improvement projects.

7. Consider using an EM organizational structure to put environmental management on an equal basis with operations and maintenance.

8. Use a team approach in staffing the pillars and coordinating their activities. Using teams would provide a laboratory for testing team building techniques and skills at AEDC, especially the techniques for team learning advocated by Peter Senge²².

9. Consider a procedure for including environmental management in every project accomplished on the installation.

10. Concentrate project efforts on Category I and II projects since they have funding priority.

11. Scope projects to maximize achievement of environmental goals and objectives.

-
- ¹ Arthur A. Thompson, Jr. and A. J. Strickland III, Strategic Management , 8th ed. (Chicago: Irwin, 1995), p. 11.
- ² Thompson and Strickland, p. 37.
- ³ Thompson and Strickland, p. 91.
- ⁴ Harold Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling , 5th ed, (New York: Van Nostrand Reinhold, 1995), pp. 605-607.
- ⁵ Glenn Harkness, Michael Marchlik, and Robert Polito, "Integrating Environmental Compliance Activities Into Total Business Plans", AIPE Facilities , 22, No. 3 (May/June 1995), p.34.
- ⁶ Daniel A. Nathanson, Robert K. Kazanjian, and Jay R. Galbraith, "Effective Strategic Planning and the Role of Organizational Design", Implementation of Strategic Planning , Peter Lorange, ed, (Englewood Cliffs, NJ: Prentice Hall, 1979), p.111.
- ⁷ Robert Springmier, "Planning as a Vehicle for Strategic Redirection in a Matrix Structure", Implementation of Strategic Planning , Peter Lorange, Ed. (Englewood Cliffs, NJ: Prentice-Hall, 1979), p. 194.
- ⁸ Robert J. Andreoli, "Organizing for Environmental Compliance", Federal Facilities Environmental Journal, 1, No. 3, (Autumn 1990), pp. 281- 287.
- ⁹ Amy Lientz and Joel Schurke, "Implementing Waste Minimization at the Idaho National Engineering Laboratory", Federal Facilities Environmental Journal , 3, No 4, (Winter 92/93), pp. 407-432.
- ¹⁰ Air Force Center for Environmental Excellence, " Hazardous Materials Management at the Naval Air Weapons Station, Point Mugu, California", Federal Facilities Environmental Journal , 4, No. 2. (Summer 1993), pp. 151-165.
- ¹¹ Sherrie W. Goodman, ""Environmental Security: Defending Our Future". Environmental Manager , 1, (March 1995), pp. 11-13.
- ¹² Major Ronald D. Marlin, "Environmental Budgeting", AFI 32-70019 (May 1994).
- ¹³ Anthony G. Bower and Peter D. Jacobson, "A DOD Strategy for Pollution Prevention", Federal Facilities Environmental Journal , 4, No. 2, (Summer 1993), pp. 185-201.
- ¹⁴ Benjamin S. Blanchard and Wolter J. Fabrycky, Systems Engineering and Analysis ,(Englewood Cliffs, NJ : Prentice Hall, 1990), p. 123.
- ¹⁵ William E. Souder, Project Selection and Economic Appraisal , (New York: Van Nostrand Reinhold, 1984), pp. 164-165.
- ¹⁶ Thompson and Strickland, p. 278.
- ¹⁷ Souder, pp. 47-48.
- ¹⁸ R. Wayne Mondy and Shane R. Premeaux, Management , (Boston: Allyn and Bacon, 1993), p. 120.
- ¹⁹ Norman Gaither, Production and Operations Management , 6th ed, (Fort Worth, TX: Dryden Press, 1994), p. 264.
- ²⁰ Kerzner, p. 883-884.
- ²¹ Thompson and Strickland, p. 110.
- ²² Peter M. Senge, The Fifth Discipline , (New York: Doubleday, 1990), p.233.

MIRTIS: A MODEL-INTEGRATED
REAL-TIME IMAGE PROCESSING SYSTEM

Michael S. Moore
PhD. Candidate
Department of Computer and Electrical Engineering

Vanderbilt University
400 24th Ave. S.
Nashville, TN 37235-1824

Final Report for:
Graduate Student Research Program
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

September 1995

MIRTIS: A MODEL-INTEGRATED
REAL-TIME IMAGE PROCESSING SYSTEM

Michael S. Moore
PhD. Candidate
Department of Electrical Engineering
Vanderbilt University

Abstract

MIRTIS (Model Integrated Real-Time Image processing System) is an environment which employs model-based synthesis techniques to generate real-time image processing applications. MIRTIS is capable of creating very high performance implementations of a large class of image processing computations. It automatically data parallelizes the computations using the split-and-merge processing model and executes them on a parallel hardware architecture, a network of C40 DSPs. During the 1995 RDL/AFOSF Graduate Student Summer Research Program, improvements were made to the MIRTIS system which will enable it to be used in tests at AEDC in the near future.

1 Introduction

1.1 Motivation

Non-dedicated image processing applications users usually have to sacrifice algorithm implementation flexibility for real-time performance. Most existing off-the-shelf real-time systems use specialized hardware architectures to perform specific algorithms in real-time (eg. convolution). A drawback of specialized hardware is that it cannot always be re-programmed with new or non-standard algorithms. Some imaging facilities, such as the one at Arnold Engineering Development Center (AEDC), need a system which can be rapidly programmed, configured, and scaled to solve a wide variety of problems.

AEDC provides technical support for aerospace tests which involve the largest and most technologically advanced flight simulation facilities in the world. Information extracted from video test data is often needed for analysis during and after the flight simulations. Since the fast availability of this information can be quite critical to the success of the tests, the data must be processed as quickly as possible. In some cases, the processing must be done in real-time. The existing hardware solutions at AEDC (a Quantex QX-7 system, and various PC based imaging boards) do not always meet their needs because of limits in capability, reliability, and algorithmic flexibility.

Through the search for a replacement for AEDC's current real-time imaging systems, it has become evident that the real-time image processing equipment currently available "off the shelf" is

not the most cost effective solution to AEDC's test support needs. Since the support requirements are constantly changing, they require a system which is rapidly programmable, configurable, and scalable. To lessen the hardware costs, the system must be modular and use as little specialized hardware as possible. To minimize training and programming costs, the system must include a very high level programming environment to provide simple and transparent access to the underlying hardware architecture.

1.2 The MIRTIS Project

This paper presents MIRTIS (Model-Integrated Real-Time Image processing System). MIRTIS is a real-time image processing environment and architecture being developed through a joint effort between Vanderbilt University and AEDC. The goal of the on-going project is to build a machine and programming environment which not only meets AEDC's needs, but also solves as general a class of problems as possible with real-time performance. In addition to AEDC's real-time video processing applications, the system should be applicable to other problem domains, such as low-level vision for robotics, autonomous navigation, tracking, remote sensing, etc. The goals of MIRTIS are as follows:

- **Scalable Real-Time Performance** to meet the requirements of a broad class of applications. The system must be easily scaled up or down for a particular application.
- **Programmability** so engineers can rapid prototype experimental/custom algorithms and create real-time computational dataflows consisting of both these and standard algorithms.
- **Flexibility** so that the system can be rapidly changed to meet the requirements of a new application, including the software and hardware configurations, and the real-time constraints.
- **High Level Parallel Programming Interface** so the programmer can write parallel programs without dealing with parallel issues. Programming style should be independent of the underlying parallel hardware architecture.
- **High Level Tools** to enable rapid reconfiguration and scaling, and to make the system easy to use.
- **Dynamic Interactivity** to enable algorithmic parameters to be adjusted "on the fly", while the system is running. Users should be able to experiment with and fine tune the algorithms to meet the application requirements.

In order to meet these goals, MIRTIS employs model-based synthesis, a method of managing complexity in large engineering systems citekarsai. Using this technique, MIRTIS automatically "synthesizes" data parallel image processing applications from high level specifications and executes them on a parallel network of Texas Instruments TMS320C40 DSPs (C40s). Real-Time execution is

achieved by scaling the data parallelism to the appropriate level to meeting the timing specifications. The scaling is done with virtually no overhead via a communications concept called Pipeline Cut-Through (PCT). Since the parallel system is automatically synthesized, the user is insulated from the complexities involved in integrating a parallel system.

The following sections discuss the implementation of MIRTIS. First, the problem domain is defined and the difficulties of building a general solution are considered. Next, the implementation details of MIRTIS are discussed, showing how model based synthesis techniques are used to hide the system complexity. An example application is discussed which demonstrates how MIRTIS will be used in testing turbine engines at AEDC.

2 Background

2.1 Image Processing Problem Domain

Image processing involves the mathematical manipulation of digital images. In the field of computer vision, image processing operations are referred to as early vision, or low-level vision. Throughout this paper, the term **image processing** refers to the image to image transformations which constitute low-level vision. Examples of image processing operations are contrast enhancement (eg. histogram stretch, histogram equalization, linear rescaling), filtering (eg. smoothing, median filter), edge detection and segmentation (eg. sobel, laplacian, Laplacian of Gaussian (LOG), thresholding), local histogram operations, and morphological operations.

2.1.1 Image Sequence Processing

The forementioned image processing algorithms operate only in the 2-D image plane. When processing image sequences, however, it is often useful to perform calculations on consecutive frames. For example, time domain averaging (averaging several frames together) is used for noise reduction, and instantaneous differencing is a simple method of motion detection. These types of algorithms are commonly used in enhancement and analysis of video data. Some techniques operate both in the 2-D image plane and in the time domain. Examples of such algorithms are 3-D filtering, and 3-D morphology. It is important for a general image processing system to have the capability of performing these types of algorithms.

2.1.2 Properties

Image processing has been the most common area for the application of high performance parallel computing [1]. The operations have several characteristics which make them particularly suitable for implementation on parallel computers [2].

- They are regular. Usually the same computation is performed for each pixel in the image.

- Most image processing programs traverse the image in raster order. This commonality in procedure makes it easy to parallelize many algorithms with the same technique.
- Usually the algorithms perform fairly simple computations to produce each output pixel.
- Images are large data sets which lend themselves easily to data parallelism.
- Processing sequences of images with these algorithms requires high computational power, and many applications (especially the real-time applications) need the speedups of parallelization.

2.1.3 Parallel Computation Models for Image Processing

There are several parallel computation models which can be applied to image processing. For instance, SIMD (Single Instruction, Multiple Data) and MIMD (Multiple Instruction, Multiple Data) architectures have been applied to image processing: SIMD architectures traditionally being used for low-level image processing, and MIMD for mid and high-level vision operations. At first look, it might seem apparent that SIMD architectures are optimal for the problem domain we have defined. SIMD solutions are intuitively applicable to image processing, since the same simple operation is performed across the image. Fine grained SIMD image processing architectures have been built which take advantage of this property. However, a closer study reveals that truly SIMD architectures suffer in scalability, programmability, and flexibility. A more practical, and no less intuitive approach is possible by using coarser grained MIMD architectures with fewer more powerful processors. With a parallel programming paradigm called split-and-merge, which is sort of a software version of SIMD computation, MIMD architectures can easily be exploited for image processing. By using a MIMD architecture with fewer more powerful processors, we gain flexibility, programmability, and ease of use.

Another way in which parallel image processing architectures can differ is the memory model used. Both *shared memory* and *distributed memory (message passing)* architectures have been used for image processing. Although shared memory systems have the advantage of being easy to program, using a shared memory model has several drawbacks. One drawback is that systems which physically share memory are *fixed* architectures, and therefore are not modular, extensible, or scalable. *Virtual shared memory* architectures, such as the KSR, actually have physically distributed memory, but emulate a shared memory space between distributed memory banks by passing messages. A drawback of virtual memory architectures is that the scalability is highly problem dependent, and moreover implementation dependent. By the same causes, it is also very difficult in general to form dependable computational performance models for such systems. These models are crucial to the development of a parallel real-time image processing system. Distributed memory/message passing architectures, although they are in general more difficult to program, can be designed to be much more scalable and flexible than shared memory machines. Therefore, we assumed a distributed memory hardware architecture when designing the MIRTIS system.

2.1.4 Split-and-Merge Data Parallelism

Because of the properties mentioned above, image processing algorithms are easily data parallelizable. A simple data parallel programming technique which is applicable to image processing is the *split-and-merge model*. In this technique, the input data is split into N pieces, which can be blocks of rows, blocks of columns, panels, etc. The pieces are distributed to several *worker* processors, each which performs the same algorithm on it's section of the data. The partial results are then merged to form the output (see figure 1).

Since each of the N processing nodes computes $\frac{1}{N}th$ of the result, there is a computational speedup of at most N . The actual speedup depends upon the communications overhead. Sources of communication overhead in the split-and-merge processing model are:

- splitting: distributing the data to the processors
- merging: combining the results to form the output
- sharing: communicating *shared* data between worker processors

The splitting and merging overheads are obvious. However, the *sharing* overhead requires some explanation. Some algorithms will require that part or all of the data be available to more than one of the worker processors. Consider, for example, a 5x5 convolution operation, which is implemented by the equation below.

$$O(i, j) = \sum_{k=-2}^2 \sum_{l=-2}^2 I(i-k, j-l) * H(k, l) \quad (1)$$

Where H is the 5x5 convolution kernel. Suppose that pixel $I(p, q)$ is a pixel lying just above a horizontal cut in the input image. In order to correctly calculate $O(p, q)$, the processor must access data from five different rows in the input image, two which are below the horizontal cut. Since those two rows have been distributed to a different processor, these rows need to be shared by two processors. If the architecture is a distributed memory multicomputer (as we are assuming), this sharing, or *overlapping* of data, must be provided via inter-processor communications¹.

In [2], Webb shows that a very large class of operations can be performed with the split-and-merge model. He proves that any operation that can be computed in forward or reverse order over a data structure can be performed invariantly with the split-and-merge technique. We will call these algorithms *splittable* computations. Examples of non-splittable computations are error-diffusion halftoning and some region-segmentation algorithms which require a particular image scanning order [2]. Since these order dependent algorithms are relatively rare in image processing, splittable image processing computations form a very general class of algorithms.

¹ A simplification of this general split-and-merge model will be made later which places restrictions on the access to the "shared" data. This will allow the formation of a predictable split-and-merge performance model.

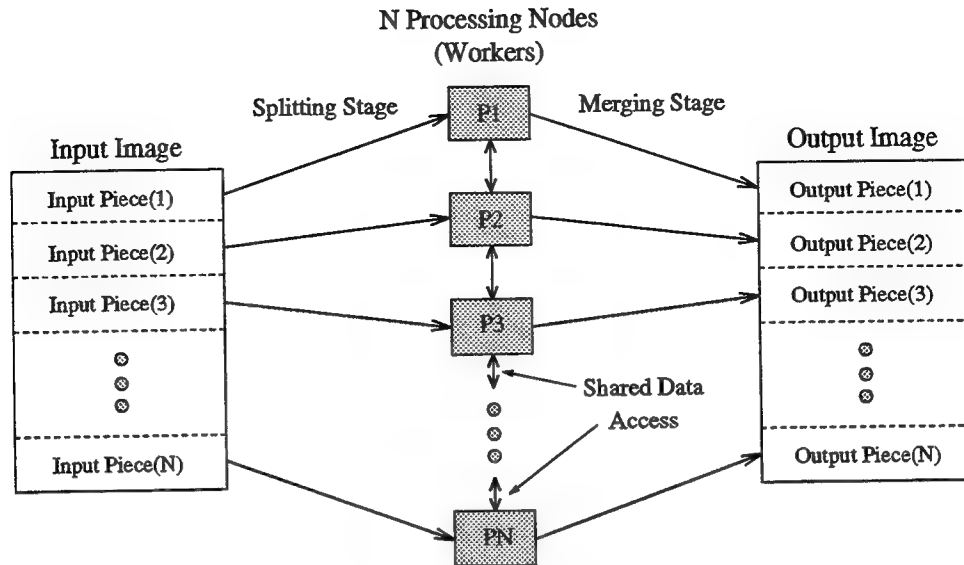


Figure 1: The Split-and-Merge Processing Model

In order to employ the split-and-merge technique in real-time imaging, several issues must be addressed. (1) Assuming a scalable distributed memory hardware architecture, can we form a performance model for a split-and-merge computation? (2) Moreover, can the computation be scaled to achieve the required execution time? (3) If so, how many processors are required? Unfortunately, the answers to these questions are dependent on the particular implementation and hardware architecture.

The difficulty in forming a performance model is that the communication overheads are not characterizable without knowledge of the implementation and architecture. Especially problematic is characterizing the data sharing overhead, since not all algorithms access data in the same way, and not all communications networks guarantee performance. However, with appropriate assumptions about the hardware architecture, a priori knowledge about the behavior of the algorithm, and a simplification of the split-and-merge processing model, the performance qualities become predictable. We make the following assumptions:

- The hardware network supports simultaneous communication and computation (communications and CPU can operate concurrently).
- We know a priori how the data is accessed by the algorithms. It is known how much of the input and output data must overlap. For example, a 5x5 convolution requires 2 rows (or columns) of overlap data along each side of a split in the input image.

These are fair assumptions, since (1) support for concurrent communication and computation is a current trend in parallel systems, and (2) due to the properties of image processing computations mentioned above, the way the data is accessed will be known a priori for almost all algorithms. The

first assumption allows the splitting and merging overheads to be neglected in the performance model because the data can then be distributed and results collected simultaneously with the computations. The second allows a simplification of the split-and-merge processing model. Instead of communicating the overlap data between worker processors, which adds uncertainty to the performance model, we can merely give each worker all the data needed to perform its task during the splitting stage. Each pixel lying in an overlap will be given to at least two processors. By making all input data *read only* (outputs go into a different data structure), there is no computational conflict.

This method of overlapping the data does increase the communications bandwidth required to split the data (when overlap is required), which reduces the maximum achievable throughput to less than the communications hardware bandwidth. The more overlap data an algorithm requires, the greater the effect of this degradation in maximum achievable throughput. However, a very large part of the commonly used image processing algorithms are *local operations*, in which each output pixel is calculated from the values within a small neighborhood of the corresponding input pixel. Local operations require small overlaps, so the potential system throughput is very close to the bandwidth of the communications network. For non-local operations, which require large overlaps in the data, the maximum computational throughput that can be achieved is more significantly less than the maximum communications bandwidth of the hardware.

This does not imply that the split-and-merge model cannot produce real-time implementations of non local operations. However, in order to achieve a throughput of X for non-local split-and-merge operations requires that the communications hardware have a bandwidth significantly more than X . In a later section, the PCT communications technique will be presented. It can be shown that in the worst case, where the entire input image is required to calculate each output pixel, the actual inter-processor communications bandwidth required by PCT to achieve a system throughput of X using the split-and-merge model is $\frac{2N-1}{N}X$, which is bounded above by $2X$.

Another way of dealing with non local operations is to split the data in the time domain instead spatially. With this variation of the split-and-merge model, called *temporal splitting*, each processor computes an entire image. There is no data sharing between processors, so the maximum achievable throughput is again the same as the communications hardware bandwidth. The draw-backs to using temporal splitting are that (1) since no data is shared, it is impossible to perform 3-D computation with temporal splitting, and that (2) the *latency*, the time it takes to compute one image, does not decrease as the number of processors increases.

2.2 Challenges of Real-Time Image Processing

A *Real-Time* system is one which, due to interaction with its environment, must produce outputs which are not only numerically correct, but also meet timing constraints. An image processing system is considered to be real-time if it obtains image data from sensors and must produce outputs

according to temporal constraints.

Performing Image processing computations in real-time is difficult. The biggest reason is that these computations are by nature slow due to the large size of image data. For example, data rates resulting from some common video digitization resolutions are given in Table 1. The last column is the corresponding computation rate required to perform a 5×5 convolution on that data in real-time. This computation rate is calculated by multiplying the pixel rate by 50, since a 5×5 convolution operation requires 25 multiplies and 25 adds for each output pixel. The computation rate is given in $\frac{Mops}{sec}$, or Million generic math operations per second, which may or may not correspond to a machine cycle, depending upon the processing node architecture. Note that even for moderate image sizes, such as 512×480 , the computational data rate required by this simple filtering operation is very high ($370 \frac{Mops}{sec}$).

Table 1: Throughput Requirements Of Real-Time Video Processing

resolution colsxrows	depth $\frac{bits}{pixel}$	frame rate $\frac{frames}{sec}$	data rate $\frac{Mbytes}{sec}$	5x5conv $\frac{Mops}{sec}$
128x128	8	30	0.47	25
256x256	8	30	1.9	98
512x480	8	30	7.0	370
640x480	8	30	8.8	460
720x486	8	30	10	524
1024x1024	8	30	30	1,600
128x128	16	30	.94	25
256x256	16	30	3.7	98
512x480	16	30	14	370
640x480	16	30	18	460
720x486	16	30	21	524
1024x1024	16	30	60	1,600

2.2.1 Hardware Architectural Issues

As is shown in Table 1, the computational power required for real time image processing far exceeds the capabilities of traditional computer architectures. Moreover, most applications require that a sequence of such algorithms be applied to the data (eg. gamma correction, then time domain averaging, then convolution, then thresholding, etc.). Performing general sequences of image processing computations on video data in real-time requires a high performance, scalable, parallel hardware architecture.

This parallel hardware architecture must be capable not only of performing the computations

in real-time, but also of supporting the huge system I/O and internal communications bandwidths necessary for real-time image processing.

- **computation speed:** As is evident in Table 1, incredible computational rates are required by real-time image processing. Very high performance processing nodes are required.
- **communication throughput:** Huge inter-processor communication rates are involved. The computational network must have a communication system capable of sustaining the data rates shown in Table 1. The communication rates necessary for real-time image processing require high performance communications hardware.
- **I/O:** The I/O interface to a digital imaging system must either be:
 - analog video – input data acquired by digitizing video with a special video A/D device and output data converted to displayable video by a D/A.
 - digital interfaces – I/O interfaces may be digital connections to other system components (real-time disks, digital cameras, other vision systems, etc.)

These types of interfaces do not often exist on modern commercially available parallel computers [1].

- **system latency:** In some applications, the system latency (the time between sensing a visual event and outputting the results from that event) is critical due to the nature of the application. For example, researchers at CMU used real-time vision for navigation in their NAVLAB experiment, in which a van moving at highway speeds used visual sensing for autonomous navigation [3] [1] [4]. The latency of the image processing system for this application is obviously constrained heavily. The hardware architecture must support low latency both in the communications and computations.

2.2.2 Software Architectural Issues

It is a well known fact that the largest obstacle to generating real world parallel applications is not building the hardware, but providing programming environments which will allow programmers to utilize it easily and efficiently. The complexities of parallel systems can easily overwhelm even an experienced programmer. Task decomposition, sub-task allocation, scheduling, inter-task communication, and synchronization are major sources of complexity in parallel systems. The main issue that must be faced in developing a real time image processing environment is how to deal with these complexities in a way that takes advantage of the properties of image processing algorithms. A suitable software environment is needed which provides the user with (1) high level methods of specifying the computations and real-time constraints, and (2) a programming paradigm with transparent access to the parallel facilities.

2.2.3 Real-Time Issues

The relevant real-time constraints for image processing are throughput and latency. In order to meet throughput and latency requirements, information is needed about the performance properties of the computations to be performed relative to the hardware architecture, the network topology, and the behavior of the communication network. Mathematical models of the system performance (throughput and latency) must be formulated in order to determine (1) the granularity of parallelism required, and (2) the mapping of the computations to the hardware necessary to meet the constraints. These models must be deterministic in order to guarantee that a particular computation can or cannot be done in real-time on a particular hardware configuration. Thus, simplicity in the parallelism scheme is quite important.

3 System Overview

This section presents a real-time image processing system. First the low-level details are given, and then the system architecture is presented, showing how the details of the underlying parallel implementation are hidden from the user.

3.1 Low-Level Details

3.1.1 Hardware Architecture

The current trends in parallel hardware technology have produced parallel processing building blocks (DSPs, etc.) powerful enough for real-time image processing. The MIRTIS hardware platform is a pipeline connected network of Texas Instruments TMS320C40 DSPs (C40s). We chose the C40 for the following reasons:

- C40s offer very high performance
 - 40Mflop processor
 - 6 bi-directional communication links, $20 \frac{Mbytes}{sec}$ each
 - 6 autonomous DMA co-processors
- We have gained experience using C40s in other projects, and have built tools for parallel C40 development [5][6].
- A large number of C40s were available to us at the time the project started.

Our current systems are made up of passive VME chassis containing up to 20 4-processor boards and PC backplanes containing up to 8 4-processor boards. The system I/O is provided by special C40 frame grabber (A/D) and display (D/A) modules. The link connections are made with normal

ribbon cable. Note that although the current MIRTIS hardware architecture is based on the C40, the system could be built from either other types of DSPs or from more general purpose processors (eg. C44, C80, T9000, SHARC, Alpha, etc.).

3.1.2 Pipeline Cut-Through Communication Scheme

Overview: Pipeline cut-through (PCT) is a communications technique used to implement the split-and-merge processing model. PCT automates the splitting and merging processes and makes the data parallelism transparent to the programmer. It splits the input image data into appropriately overlapping pieces and communicates the pieces to the worker processing nodes. The image processing program running on each node receives its local input and output buffers as function parameters and performs the computation. During the computation, the PCT system splits the next input image and again distributes it across the network. After the computation has finished, the PCT system simultaneously merges the partial results and restarts the computation. Since the communication is concurrent to the computation, each node is very nearly always computing.

Implementation: PCT has been implemented on a pipeline connected network of C40 DSPs. The implementation uses the autonomous DMA co-processors of the C40. Each DMA may transfer data from memory to memory, from port to memory, or from memory to port without CPU intervention. Since the C40 communication ports are memory mapped devices, a DMA can also be programmed to transfer from port to port by writing to (reading from) the port address. With a DMA programmed to transfer from port to port, the C40 is effectively a link (wire) connecting its two neighbors. (This is why a pipeline topology works, as will be seen.) The DMA engines have a special feature which allows them to auto-initialize themselves after a transfer terminates. Thus, they can be programmed to perform several tasks, even a circular-linked list of tasks, independently of the CPU (see "Fun With Link Pointers" [7]).

PCT uses three basic communication *modes*: (1) **receive**: read input data from input port to memory, (2) **send**: send results from memory to output port, and (3) **forward**: send data from input port to output port (act like a wire). By taking advantage of link pointer autoinitialization, the DMAs can be programmed with a "communications state-table". These state-tables are designed to make a group of processors perform as a split-and-merge communication group. The state-table must program the DMAs with the correctly timed sequences of receive, send, and forward that cause the input stream to be distributed across the memories of the group, and the output stream to be assembled.

The example in Figure 2 shows a 3 node PCT split-and-merge group. The image is being split into 3 blocks of rows and distributed across the memories of the 3 processing elements. Simultaneously, results are being read from the memories and merged to form the output. At any time, exactly one of the processors is concurrently receiving and sending data. All other processors in that group are

forwarding. Effectively, what happens is that the processors take turns being connected to the input and output stream.

Note that in the example in Figure 2 two simplifications have been made to make the figure understandable. (1) The processor which is receiving and sending is not shown to be processing at the same time. The double buffering of the input and output data which allows this to happen are not shown. (2) No data is being shared by the processors. The implementation of the data sharing requires additional communication states in which processors insert extra copies of the data into the stream at the appropriate time during the splitting process. This insertion process is quite complicated, and for that reason the details of how it works are not discussed here.

Programming the DMA co-processors to achieve the appropriate communication patterns is tedious. The DMA state table is programmed based on several variables which are set at run-time, including the image dimensions, the overlap data required by the algorithm being executed, the number of processors in the split-and-merge group, the local input and output port numbers, and the processor's location in the group. These variable are used to determine (1) the amount of data to be received to and sent from local memory (2) the amount of data to be forwarded, and (3) how much and when overlap data must be inserted into the stream. Using normal programming methods to build these PCT programs is not feasible. In order to make the system usable, it is necessary to somehow automatically generate the PCT communications programs. MIRTIS uses model-based synthesis techniques to synthesizing real-time PCT applications.

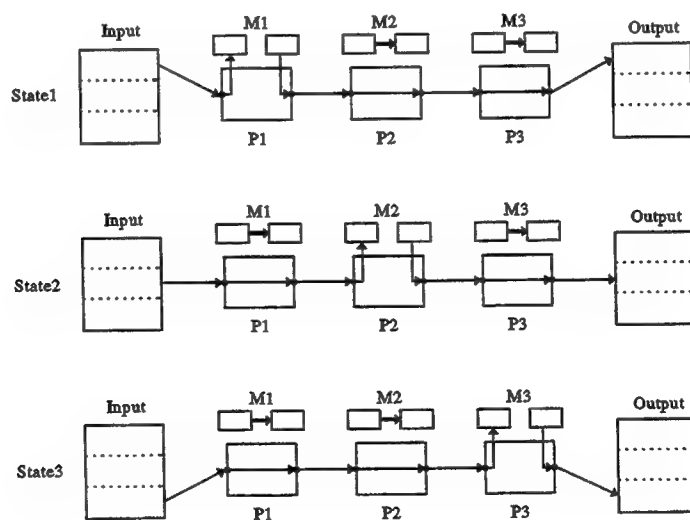


Figure 2: Pipeline Cut-Through Scheme

3.2 Model Integrated Program Synthesis

Model Integrated Program Synthesis (MIPS) is a method of managing complexity in large scale engineering systems. The system is generated from *models*, which specify the application in terms of a paradigm, or language. Model interpreters *compile* the models and synthesize an implementation. The Multigraph Architecture (MGA) is a MIPS architecture developed at Vanderbilt which provides a frame-work and tools for (1) building graphical domain specific models and (2) transforming the graphical models into executable applications [8]. By using domain specific models and interpreters, MGA allows the domain experts to specify a system in familiar terms without dealing with the underlying software engineering details.

4 The MIRTIS Architecture

We have used the MGA to create MIRTIS, an environment which automatically generates the complex PCT split-and-merge DMA programs and synthesizes real-time image processing computations. An image processing modeling paradigm has been developed which contains the image processing concepts necessary to automatically build PCT split-and-merge dataflows for general image processing algorithms. The paradigm includes aspects for specification of the image processing computations to be performed, the hardware architecture to be used, and the real-time constraints which must be met. Based on these models, the model interpreter automatically generates the necessary PCT communications programs and computation schedules to implement the data-parallel computational dataflow. It then maps this dataflow to the parallel hardware architecture. The details of the parallel implementation are completely hidden from the user.

The MIRTIS system architecture follows the basic model-based system architecture shown in [9] and [11]. It consists of the following components (see Figure 3):

- A Modeling Environment
- A Model Database
- The MIRTIS Model Interpreter
- An Image Processing Application Library
- The PCT Run-Time Library
- The MIRTIS Graphical User Interface
- A Network of C40s

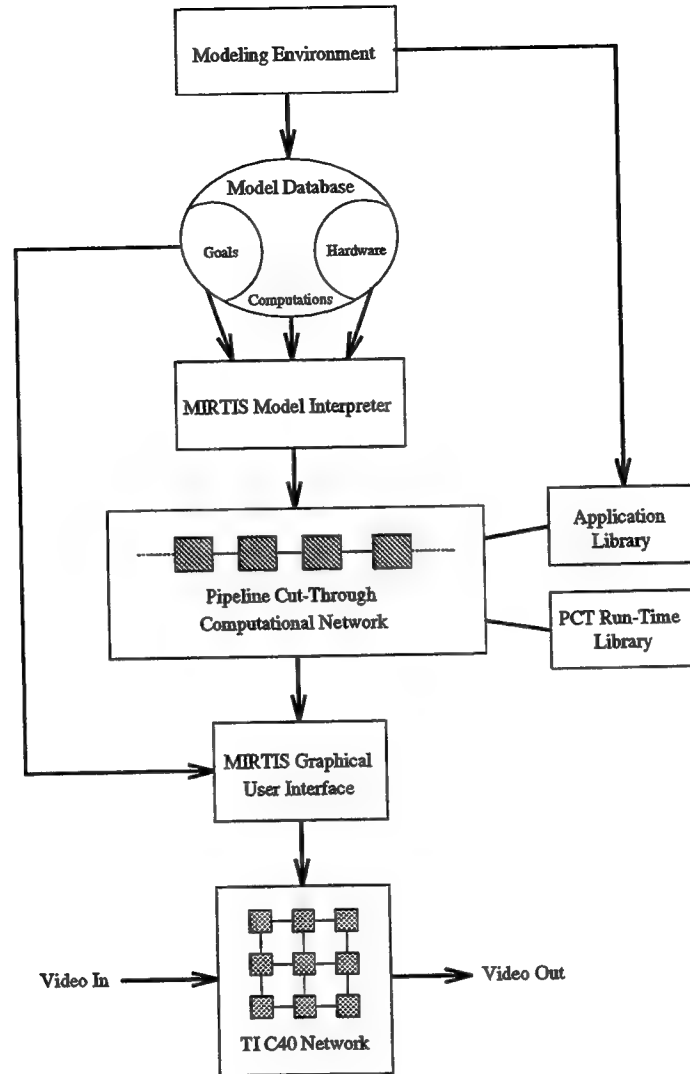


Figure 3: The MIRTIS Architecture

4.1 The Real-Time Image Processing Modeling Paradigm

A modeling paradigm has been developed for real-time image processing. It includes three modeling views, or *aspects*: (1) the *Computation aspect*, (2) the *Hardware aspect*, and (3) the *Goals aspect*.

4.1.1 Computation Aspect

The computation aspect describes which algorithms are to be performed, and in what order. The computation model is represented visually in a graph dataflow style: blocks representing computations and inter-connecting lines representing the flow of data. The computation aspect includes two types of models: (1) Library Function Models and (2) Application Models.

Function models describe the individual algorithms available in the image processing application library. Functions are the only models with which actual image processing code is associated, and are thus the elementary blocks which eventually implement the dataflow. A function model contains pertinent information about the library function with which it is associated, such as its performance behavior, the overlap data it requires, etc. An **application model** is a dataflow consisting of function models and their interconnections. This is the top level dataflow model.

4.1.2 Hardware Aspect

The hardware aspect model describes the C40 network, including information about each node (how much memory, node type) and about the interconnection topology. This model is automatically acquired using a C40 network utility tool called *TICK* [5]. (*TICK* also provides the network loading and debugging.) The only requirement of the C40 network is that it include some path from the input node (a C40 digitizer) to the output node (a C40 display module).

4.1.3 Goals Aspect

The goals aspect allows specification of formal computational performance goals to the model interpreter. These goals are used in making decisions about the type and level of parallelism which will be used. The goals models contain: (1) a *Throughput Goal*, (2) a *Latency Goal*, and (3) a *Failure mode*. The throughput and latency goals quantify performance requirements of the application. The failure mode specifies what the interpreter should do if the goals cannot be, or are not being met. The possible actions are to report the problem, down-sample the input data, or merely fail. For example, in some latency critical applications, if the hardware is not capable of supporting the requested latency, the system must give a warning and fail. However, in other less critical applications, down-sampling the data would be perfectly acceptable.

4.2 The Modeling Environment and Model Database

A modeling environment facilitates the creation of system models. The MIRTIS *graphical model editor* is used for building and manipulating the system models. The models represent the system in terms of the MIRTIS modeling paradigm discussed earlier. The model editor is based on **XVPE**, a configurable visual programming environment developed at Vanderbilt University [8] which runs under **X** and uses the **obst** object oriented database package for model storage. **XVPE** has been ported to **linux**, which makes a PC a convenient platform for modeling. A *graphical user interface* is used to control the operation of the model interpreter, to load the C40 network, and to monitor the performance of the running system. A *library code development tool* is used to maintain and develop the image processing application library.

4.3 The MIRTIS Model Interpreter

The MIRTIS model interpreter, called *MINT*, interfaces with the graphical model builder and automatically generates a PCT computational network which realizes the computations described in the models. During the interpretation process, the interpreter (1) analyzes the graphical system models (2) partitions the computation model (3) determines the appropriate numbers of processors to be used for each of the sub-computations (4) parallelizes the sub-computations (5) maps the parallelized sub-computations to the hardware network, and then (6) produces a pipeline cut-through computation network which realized the computational dataflow with the specified performance.

4.4 The Image Processing Application Library

This component of the system contains the code which actually implements the image processing algorithms. Library routines are written in C, and are compiled with the standard cl30 compiler released by TI. Users can add functionality to the application library by providing C coded sub-routines. Writing these routines requires no knowledge of parallel programming, since there is no reference to parallelism in the code.

4.5 The PCT Run-Time Support Library

All parallel facilities are provided through calls to the PCT run-time support library. The library includes a scheduler, a memory manager, a dynamic setup manager, and a connection/communication manager. The main processing loop performs the inter-processor communications, dynamically re-configures the algorithmic parameters based on commands received from the user interface, runs the local computation schedule, and synchronizes the computation and communication processes. The scheduler calls the specified image processing library functions, passing to each function pointers to its context structure and its I/O data buffers. In this way, the complexity of the run-time system is not apparent in the image processing specific code, and the programmer is insulated from the difficulties of parallel programming.

4.6 The Graphical User Interface

The *graphical user interface*, called *MUI*, is used to control the operation of the model interpreter, load the C40 network, and monitor/interact with the running system. MUI allows the user to adjust algorithmic parameters dynamically. For example, the elements of a convolution kernel can be adjusted with graphical widgets while the system is running. Since the changes in the computation are immediately apparent in the output data, the user can experiment with and "tweak" algorithms easily. The ability for dynamic interaction is a very powerful feature of MIRTIS which can save time, effort, and expense in converging on a computational solution.

5 An Example MIRTIS Application at AEDC

In analyzing and testing turbine engines, information about the physical phenomena occurring within the internal the components of the running engine is needed in order to evaluate the performance, health, reliability, and efficiency of the engine. Non-destructively instrumenting a turbine engine, however, is difficult. While measuring structural stresses with strain guages has been successfully practiced, until recently there was no method of measuring fluid pressure patterns on the moving turbine blades.

Recent efforts have begun using *Pressure Sensitive Paint* (PSP) to effectively instrument the engine to obtain this valuable data. Pressure sensitive paint (actually a misnomer - it should be called oxygen sensitive paint) is a flourescent paint which, when excited with the correct wavelength of light, luminesces with a varying intensity which is functionally dependent on the oxygen depletion at the paint's surface. (The luminescence is also somewhat temperature dependent.) By (1) applying PSP to the turbine blades of jet engine's compressor stage, (2) exciting the paint with a tuned wavelength light source, and (3) correctly strobing a CCD array camera, visual data representing the pressure field on the blade can be obtained.

Such tests have already been performed using a highly light sensitive, 16-bit camera. However, since it takes at least 8 seconds to down-load one frame from the camera, they are incapable of acquiring the data quickly enough to observe the dynamics of the pressure field.

It has been proposed to use the MIRTIS system with a high quality camera to acquire and process dynamic PSP data in real-time. Figure 4 shows the proposed test setup. Preliminary tests have shown that data acquired with a 10-bit digitization resolution will exhibit significant visual variation with relavent pressure variation.

This summer, MIRTIS was instrumented with 10-bit image digitizing hardware (a Coreco F64-C40 with the 10-bit digitizer upgrade). The MIRTIS software must still be modified to work with 10-bit data. When these modifications have been completed, further experiments with MIRTIS in PSP tests promise to yield valuable dynamic pressure field data which has never before been available.

6 Summary

MIRTIS is a model-based environment for real-time image processing. The system is capable of creating very high performance implementations of a large class of image processing computations through use of the split-and-merge processing model. It automatically parallelizes the computations and executes them on a parallel hardware architecture, a network of C40 DSPs. MIRTIS is completely programmable, and provides a high level programming interface which masks the complexities of the underlying parallel implementation.

The current MIRTIS prototype is capable of performing dataflows of local image processing al-

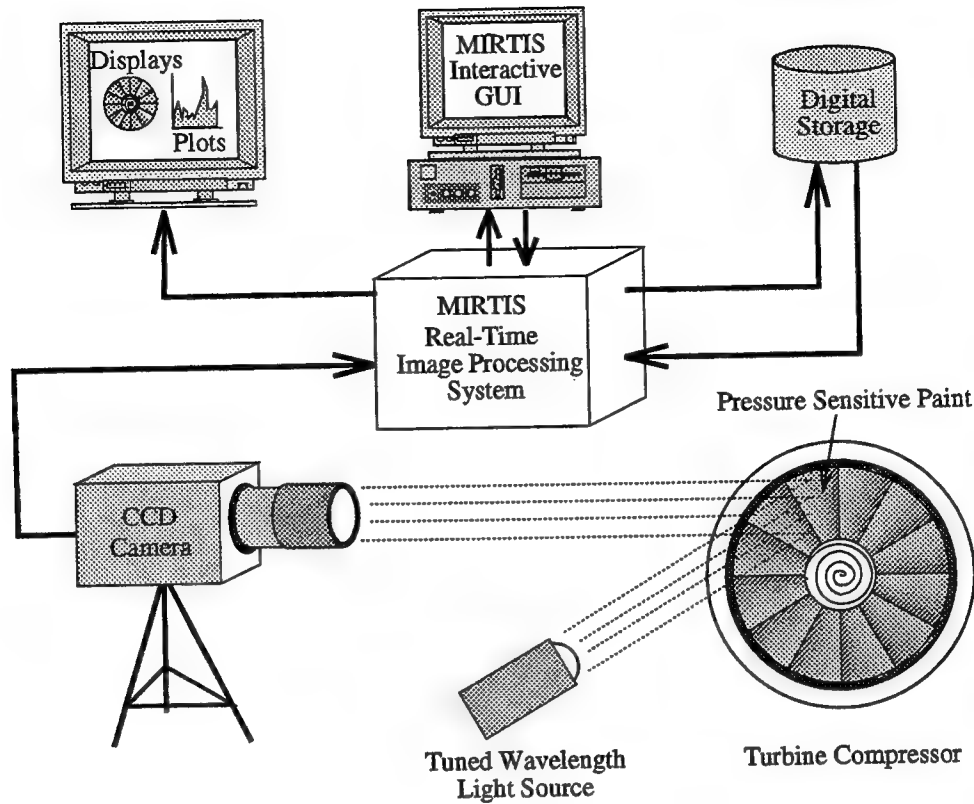


Figure 4: An Example Application of MIRTIS at AEDC

gorithms on 512×480 , $8 \frac{\text{bits}}{\text{pixel}}$ (256 grey levels) frames at video rate (30 frames per second), given that enough C40s are available in the network. The digitization resolution (width and height of the digitized frames) is adjustable. However, $7 \frac{\text{Mbytes}}{\text{sec}}$ is the highest data rate yet achieved due to the limitations of the C40 communication link (using ribbon cables reduces the C40 link bandwidth significantly). By doubly linking the hardware pipeline, though, an increase to $14 \frac{\text{Mbytes}}{\text{sec}}$ is expected. A prototype MIRTIS system containing 51 nodes has been bench-marked at $520 \frac{\text{Mops}}{\text{sec}}$ sustained (counting only useful computations) while performing a complex edge detection on live video. Systems as small as 4-6 processors have been used for simpler applications.

Future work will extend the capabilities of the system to the more general domains of medium and high level vision. The long-term goal is to build a unified architecture and programming environment for image processing and computer vision which supports heterogeneous hardware architectures and mixed modes of parallelism.

References:

- [1] Webb, John A., "High Performance Computing in Image Processing and Computer Vision", Proceedings of the International Conference on Pattern Recognition, October 1994, Jerusalem.

- [2] Webb, John A., "Steps Toward Architecture-Independent Image Processing," IEEE Computer, February 1992: p. 21-31.
- [3] Webb, John A., "Latency and Bandwidth Considerations in Parallel Robotics Image Processing," Supercomputing '93, 1993, Portland, OR: IEEE Computer Society.
- [4] Crisman, J.D. and J.A. Webb, "The Warp Machine on Navlab," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991. 13(5): p. 451-65.
- [5] B. Abbott and A. Ledeczi: "TICK: TI TMS320C40 Utility Program", Proceedings of the International Conference on Signal Processing Applications and Technology, Oct. 1994.
- [6] Ledeczi, A, Abbott, B: "Parallel Architectures With Flexible Topology", Proceedings of the Scalable High Performance Computing Conference, pp. 271-276, May 1994.
- [7] Texas Instruments: "TMS320C4x User's Guide", Texas Instruments, May 1991.
- [8] G. Karsai: "A Configurable Visual Programming Environment", IEEE Computer, March 1995.
- [9] B. Abbott, T. Bapty, C. Biegl, G. Karsai, and J. Sztipanovits: "Model-Based Software Synthesis", IEEE Software, May 1993.
- [10] M. S. Moore, G. Karsai, and J. Sztipanovits: "Model-Based Programming for Parallel Image Processing", Proceedings of the First IEEE International Conference on Image Processing (ICIP94), Nov. 1994.
- [11] M. S. Moore: "A DSP-Based Real-Time Image Processing System", Proceedings of the 6th International Conference on Signal Processing Applications and Technology (ICSPAT95), Boston, Ma., Oct. 24-26, 1995.
- [12] M. S. Moore, Nichols, J: "Model-Based Synthesis of a Real-Time Image Processing System", Proceedings of the First IEEE International Conference on Engineering of Complex Computer Systems (ICECCS95), Ft. Lauderdale, Fla., Nov. 6-10, 1995.
- [13] M. S. Moore: "A Model Based Real-Time Image Processing System", Final report for the 1994 USAF-RDL summer research program, AFOSR Contract F49620-93-C-0063, September 1994.
- [14] M. S. Moore: "A Programming Environment for Parallel Image Processing: Progress Report", Report Prepared for Sverdrup Technology, Inc., 31 January 1993.
- [15] M. S. Moore: "A Programming Environment for Parallel Image Processing: Progress Report 2," Report Prepared for Sverdrup Technology, Inc., 31 March, 1993.
- [16] P. A. Laplante, "Issues in Real-Time Image Processing," Proceedings of the 1993 IEEE Systems, Man, and Cybernetics Conference, Le Touquet, France, October, 1993, pp. 323-326.

**NONUNIFORM UNSTEADY BOUNDARY CONDITIONS
FOR THE EULER EQUATIONS**

**Thomas H. Olsen
Graduate Student
Department of Mechanical and Aeronautical Engineering**

**University of California, Davis
Davis, California 95616**

**Final Report for:
Graduate Student Research Program
Arnold Engineering and Development Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Arnold Engineering and Development Center**

September 1995

NONUNIFORM UNSTEADY BOUNDARY CONDITIONS FOR THE EULER EQUATIONS

Thomas H. Olsen
Graduate Student
Department of Mechanical and Aeronautical Engineering
University of California, Davis

Abstract

Computational fluid dynamics (CFD) is currently being used to study a variety of unsteady aerodynamic problems. Some examples of these problems are store separation, high angle of attack aerodynamics, and the flow through engine components. These problems require the use of outflow boundary conditions which properly convect the flow out of the computational domain. In order to use CFD to study the effect of atmospheric winds on vehicle aerodynamics and flight dynamics and controls, current far-field boundary conditions must be modified to allow for an unsteady inflow boundary.

In this paper, nonuniform unsteady boundary conditions are developed for use with the Euler Equations. These boundary conditions are developed to allow for the specification of a flow, at an outer boundary, which can vary in both space and time. These boundary conditions are extensions of nonreflecting boundary conditions currently being used.

NONUNIFORM UNSTEADY BOUNDARY CONDITIONS FOR THE EULER EQUATIONS

Thomas H. Olsen

Introduction

The numerical solution of the Euler equations, on a finite domain, requires some knowledge of the flow on the outer boundaries. Generally, a steady flow is assumed at these boundaries (see, for example, [9, 16, 7]). Methods which account for an unsteady outflow condition also exist (see [4, 13, 11, 5]).

Although a steady inflow boundary condition is sufficient for the majority of aerodynamic applications, several problems require a nonuniform unsteady inflow boundary condition. The flight of an aircraft through a microburst is a simple example. The microburst is an unsteady gust of wind which could be modeled within the computational domain. However, to reduce the cost of the computation, the microburst should be allowed to enter the domain through a far-field boundary.

Another useful application of an unsteady inflow boundary condition is the problem of store separation. Currently, for store separation problems, the time-accurate solution about the complete aircraft and store is computed. This type of computation is extremely expensive. A good approximation of the problem could be obtained using the steady flow field about the complete aircraft. In this scenario, the solution would only be obtained for the store. The store will feel the effect of the aircraft through the boundary conditions which will vary according to the instantaneous location of the store relative to the aircraft. Although the effect of the store on the aircraft flow field is not accounted for in this procedure, a good preliminary estimation of the trajectory could be obtained at a substantial savings in both time and computational cost.

Further examples of problems which could utilize a nonuniform unsteady boundary condition are internal flows. A procedure similar to that described for store separation could be used to examine the various components of a jet engine. Also, for wind tunnels with slotted test section walls, porosity boundary conditions do not adequately model the flow [8]. The slotted walls could be accounted for using a nonuniform boundary condition. This would result in a solution which more accurately represents the results of a wind tunnel experiment.

This report will begin with a brief discussion on current boundary conditions. The extension of some of these boundary conditions to account for a nonuniform and, possibly, unsteady boundary will also be presented. Test cases to validate these boundary conditions will then be proposed.

Background

Current boundary conditions are generally based on one-dimensional characteristics. Starting with the one-dimensional Euler equations;

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = 0 \quad (1)$$

where

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix}, \quad E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (\rho e + p)u \end{pmatrix},$$

and the pressure is related to the conservative flow variables, Q , by the equation of state

$$p = (\gamma - 1)\rho \left[e - \frac{1}{2}u^2 \right]. \quad (2)$$

To examine the classification of the Euler equations, equation (1) is rewritten as

$$\frac{\partial Q}{\partial t} + A \frac{\partial Q}{\partial x} = 0 \quad (3)$$

where $A = \frac{\partial E}{\partial Q}$. The eigenvalues of A are u , $u + a$, and $u - a$ where u is the velocity and a is the speed of sound. Since the eigenvalues of A are real and the system has a complete set of eigenvectors, the Euler equations are hyperbolic [6]. By applying the eigensystem decomposition to equation (1), the characteristic equations can be obtained. The characteristic equations can be written as

$$T^{-1} \frac{\partial Q}{\partial t} + \Lambda T^{-1} \frac{\partial Q}{\partial x} = 0 \quad (4)$$

where Λ is the diagonal matrix of the eigenvalues of A and T^{-1} is the matrix whose rows are the left eigenvectors of A . Hence, the solution of the Euler equations propagates along the characteristic directions u , $u + a$, and $u - a$. Therefore, for a subsonic inflow we have two incoming waves and one outgoing wave. This means that we require two boundary conditions at a subsonic inflow boundary.

When a central difference scheme is used to solve the Euler equations, all of the flow variables must be specified at a boundary. Therefore, in the preceding example of subsonic inflow, the characteristic corresponding to the outgoing wave must be updated from the interior of the solution domain. This procedure is commonly referred to as the numerical treatment of the boundaries.

The most common characteristic boundary condition employs the Riemann invariants [5]. The Riemann invariants are given as

$$R^+ = v_n + \frac{2a}{\gamma-1} = \text{constant along the path } \vec{v} + a\vec{l}_n$$

and $R^- = v_n - \frac{2a}{\gamma-1} = \text{constant along the path } \vec{v} - a\vec{l}_n$.

The additional information required for the solution is generally obtained from the entropy and tangential velocity [11]. Although the Riemann Invariants are rigorously derived, they do not give adequate results for boundaries where the grid line normal to the boundary is not aligned with the streamlines.

For many flows, only certain variables are known, or certain quantities must be fixed. For example, for nozzle problems only the exit pressure might be known, or a certain mass flow rate is required at the inlet. Yee [15] has shown that care must be taken in order to maintain a stable scheme when quantities other than the characteristic variables are specified. Even though a stable scheme might still be obtained, the convergence of a given scheme might be degraded due to spurious reflections of the outgoing waves at the boundaries.

If the flow exiting the domain is inherently unsteady, care must also be taken at the boundaries to avoid degrading the accuracy of the solution. Giles [4] developed an unsteady nonreflecting boundary condition based on ΔQ . This simplifies the specification of the characteristic variables since, assuming incoming waves are steady, $\Delta W = 0$, where W is the characteristic variable.

An unsteady, nonuniform and nonlinear boundary condition has been developed by Thompson [13, 14]. This boundary condition solves the characteristic equation (4). The boundary condition is set by defining the quantity $\mathcal{L} \equiv \Lambda T^{-1} \frac{\partial Q}{\partial x}$. For incoming waves, $\mathcal{L} = 0$ which implies that the magnitude of the amplitude of the incoming waves is constant. The outgoing waves, on the other hand, are obtained from the definition of \mathcal{L} using one-sided differences. Equation (4) is then rewritten as

$$\frac{\partial Q}{\partial t} + T\mathcal{L} = 0 \quad (5)$$

and integrated in time to update the conservative variables. For multiple dimensions, the remaining derivatives are simply added to equation (5). These additional derivatives are then evaluated using central differences.

The boundary conditions discussed so far do an adequate job when the waves are aligned with the coordinate direction. However, in most real problems, the waves propagate at an angle to the grid lines. Therefore, several multi-dimensional boundary conditions have been developed. Giles has developed a two-dimensional [4] and a quasi-three-dimensional [12] boundary condition based on a Fourier analysis of the linearized Euler equations. Bayliss and Turkel [2] used an asymptotic analysis of the linearized equations for large distances to develop a nonreflecting boundary condition. A similar boundary condition has been developed by Roe [11] using the concept of bicharacteristics. These last two boundary conditions have been developed for both two- and three-dimensional problems.

The problems for which these boundary conditions are intended will involve atmospheric winds. These winds will be assumed to be aligned with the axis of the grid. Therefore, the boundary conditions based on one-dimensional characteristics will be extended to account for a nonuniform and unsteady inflow.

First, it is necessary to show that the three-dimensional Euler equations in generalized coordinates are hyperbolic. The three-dimensional Euler equations can be expressed in general curvilinear coordinates as

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \zeta} = 0 \quad (6)$$

where

$$\hat{Q} = J^{-1} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix}, \quad \hat{E} = J^{-1} \begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (\rho e + p)U - \xi_t p \end{pmatrix},$$

$$\hat{F} = J^{-1} \begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (\rho e + p)V - \eta_t p \end{pmatrix}, \quad \text{and} \quad \hat{G} = J^{-1} \begin{pmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (\rho e + p)W - \zeta_t p \end{pmatrix}.$$

U , V , and W are contravariant velocities defined as

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w, \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w, \\ \text{and } W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w. \end{aligned}$$

The pressure is related to the conservative flow variables, \hat{Q} , by the equation of state

$$p = (\gamma - 1)\rho \left[e - \frac{1}{2}(u^2 + v^2 + w^2) \right]. \quad (7)$$

The metric terms are defines as

$$\begin{aligned} \xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta), & \eta_x &= J(y_\zeta z_\xi - y_\xi z_\zeta), \\ \xi_y &= J(z_\eta x_\zeta - z_\zeta x_\eta), & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi), \\ \xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta), & \eta_z &= J(y_\xi x_\zeta - y_\zeta x_\xi), \\ \zeta_x &= J(y_\xi z_\eta - y_\eta z_\xi), & \xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z, \\ \zeta_y &= J(z_\xi x_\eta - z_\eta x_\xi), & \eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z, \\ \zeta_z &= J(x_\xi y_\eta - x_\eta y_\xi), & \zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z. \end{aligned}$$

Here J is the transformation Jacobian given by

$$J^{-1} = x_\xi (y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta (y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta (y_\xi z_\eta - y_\eta z_\xi).$$

As with the one-dimensional case, equation (6) may be written as

$$\frac{\partial \hat{Q}}{\partial \tau} + \hat{A} \frac{\partial \hat{Q}}{\partial \xi} + \hat{B} \frac{\partial \hat{Q}}{\partial \eta} + \hat{C} \frac{\partial \hat{Q}}{\partial \zeta} = 0 \quad (8)$$

where $\hat{A} = \frac{\partial \hat{E}}{\partial \hat{Q}}$, $\hat{B} = \frac{\partial \hat{F}}{\partial \hat{Q}}$, and $\hat{C} = \frac{\partial \hat{G}}{\partial \hat{Q}}$ are the flux Jacobian matrices (which are given in the appendix). The Euler equations written in general form still have real eigenvalues and a complete set of eigenvectors. Therefore, the transformation did not alter the numerical character of the equations, which are still hyperbolic. Applying the eigensystem decomposition (given in the appendix), equation (8) may be rewritten as

$$\frac{\partial \hat{Q}}{\partial \tau} + T_\xi \Lambda_\xi T_\xi^{-1} \frac{\partial \hat{Q}}{\partial \xi} + T_\eta \Lambda_\eta T_\eta^{-1} \frac{\partial \hat{Q}}{\partial \eta} + T_\zeta \Lambda_\zeta T_\zeta^{-1} \frac{\partial \hat{Q}}{\partial \zeta} = 0. \quad (9)$$

Now, knowing the type of the equations, the boundary conditions will be developed.

Proposed Boundary Conditions

The simplest boundary condition to implement is the characteristic condition based on Riemann invariants. The first step is to update the flow variables on the boundary to the new time level. Then, the characteristic boundary condition is applied in the same way as for the steady case. For subsonic inflow,

the Riemann invariant R^- , the tangential velocity v_t , and the entropy s are specified while the Riemann Invariant R^+ is extrapolated from the interior of the domain. For a subsonic outflow boundary, the Riemann invariant R^- is specified while the Riemann Invariant R^+ , the tangential velocity v_t , and the entropy s is extrapolated from the interior of the domain. Then the flow variables are obtained as follows;

$$\begin{aligned} v_n &= 0.5 (R^+ + R^-) \\ c &= \frac{1}{4(\gamma - 1)} (R^+ - R^-) \\ \rho &= \left(\frac{c^2}{s} \right)^{\frac{1}{\gamma-1}} \\ p &= \frac{c^2 \rho}{\gamma} \end{aligned}$$

Once these quantities are known, the conservative flow variables can easily be obtained.

The second proposed boundary condition is to use the idea of time-dependent characteristics as proposed by Yee [15]. The characteristic variables, W , are related to the conservative flow variables, \hat{Q} as

$$W(t) = T_{\kappa}^{-1} \hat{Q}(t)$$

where T_{κ}^{-1} is the matrix whose columns are the left eigenvectors of the flux Jacobian matrix. The subscript κ refers to ξ , η , or ζ corresponding to the flux Jacobian matrix \hat{A} , \hat{B} , or \hat{C} respectively. The characteristics entering the domain are specified while those characteristics leaving the domain are extrapolated from the interior.

Once the characteristic variables are set on the boundary, the conservative variables may be obtained from the relation

$$\hat{Q}^{n+1} = T_{\kappa} W^*$$

where the $*$ denotes the time level of the characteristic. The incoming characteristics are evaluated at time level $n+1$, while outgoing characteristics are evaluated at time level n . This might degrade the time accuracy at the boundaries. A further extrapolation, of the characteristics leaving the domain, in time might improve the time-accuracy at the boundary. It is important to note that the matrices T_{κ}^{-1} and T_{κ} are functions of the conservative variables. Therefore, an additional question of what time level to evaluate the matrices T_{κ}^{-1} and T_{κ} must be answered.

An improvement on the previous boundary condition is to use the perturbation of the characteristic variables in time to apply the boundary condition. Again, the characteristic variables are related to the flow variables by

$$\Delta W = T_{\kappa}^{-1} \Delta \hat{Q}.$$

As stated in the previous section, for a steady flow, $\Delta W = 0$ for incoming waves. For an unsteady flow, ΔW is obtained from the relation given above and the knowledge of the flow entering the computational domain. The characteristics corresponding to outgoing waves are obtained by extrapolation from the interior of the computational domain. Once ΔW is obtained on the boundary, the conservative flow variables are determined as follows

$$\begin{aligned} \Delta \hat{Q} &= T_{\kappa} \Delta W \\ \hat{Q}^{n+1} &= \hat{Q}^n + \Delta \hat{Q}. \end{aligned}$$

The next level in one-dimensional characteristic boundary conditions is to use the unsteady nonlinear boundary conditions (see [13, 14, 10]) These boundary conditions are developed from the Euler equations (6) and the eigensystem decomposition (9). For a ξ -boundary, the ξ -derivatives are written in characteristic form as

$$\frac{\partial \hat{Q}}{\partial \tau} + T_\xi \Lambda_\xi T_\xi^{-1} \frac{\partial \hat{Q}}{\partial \xi} + \hat{B} \frac{\partial \hat{Q}}{\partial \eta} + \hat{C} \frac{\partial \hat{Q}}{\partial \zeta} = 0. \quad (10)$$

Therefore, the characteristic equation in the ξ -direction is given by

$$T_\xi^{-1} \frac{\partial \hat{Q}}{\partial \tau} + \Lambda_\xi T_\xi^{-1} \frac{\partial \hat{Q}}{\partial \xi} + T_\xi^{-1} \hat{B} \frac{\partial \hat{Q}}{\partial \eta} + T_\xi^{-1} \hat{C} \frac{\partial \hat{Q}}{\partial \zeta} = 0. \quad (11)$$

Now, define $\mathcal{L} \equiv \Lambda_\xi T_\xi^{-1} \frac{\partial \hat{Q}}{\partial \xi}$. Therefore, equation (11) becomes (multiplying through by T_ξ)

$$\frac{\partial \hat{Q}}{\partial \tau} + T_\xi \mathcal{L} + \hat{B} \frac{\partial \hat{Q}}{\partial \eta} + \hat{C} \frac{\partial \hat{Q}}{\partial \zeta} = 0. \quad (12)$$

The boundary conditions can be defined in terms of \mathcal{L} . For the characteristics entering the domain, \mathcal{L} is specified. While for characteristics leaving the computational domain, \mathcal{L} is computed according to it's definition given above.

There are a couple of different possibilities for imposing an unsteady boundary condition using equation (12). The first is simply to impose the boundary condition using the definition of \mathcal{L} as

$$\mathcal{L} = \begin{cases} \Lambda_y T_y^{-1} \frac{\partial \hat{Q}_o}{\partial y}, & \text{for incoming waves} \\ \Lambda_y T_y^{-1} \frac{\partial \hat{Q}}{\partial y}, & \text{for outgoing waves} \end{cases}$$

where \hat{Q}_o is the known gust. The derivatives are evaluated using the appropriate one-sided difference formula. For the incoming waves, phantom mesh points will have to be generated outside the computational domain. An alternative method is to replace \hat{Q} in equation (11) with $\delta \hat{Q} = \hat{Q} - \hat{Q}_o$. Now, \mathcal{L} can be set as follows

$$\mathcal{L} = \begin{cases} 0, & \text{for incoming waves} \\ \Lambda_y T_y^{-1} \frac{\partial \delta \hat{Q}}{\partial y}, & \text{for outgoing waves} \end{cases}$$

Regardless of how the unsteady boundary condition is imposed, equation (11) is solved in the same manner. First, \mathcal{L} is formed according to the definitions given above. Then the derivatives tangential to the boundary (η and ζ derivatives for this example) are evaluated using central differences. Finally, the equation is then integrated in time using a second order four stage Runge-Kutta scheme.

Validation

In order to verify that the proposed boundary conditions can adequately handle the specification of a nonuniform unsteady boundary condition, a variety of test cases will be proposed. The first test case is a check to ensure that the code can still maintain freestream. This check is necessary to ensure that the boundary conditions are formulated properly and that there are no errors in the coding.

The next step will be to impose a variety of prescribed gusts. The gusts have been chosen to test some concepts which will be very important in future applications. The following gusts will be used:

1. Uniform but unsteady vertical gust.

$$w = A \sin(\omega t)$$

$$p = -A\rho\omega(z - z_o) \cos(\omega t)$$

$$\omega = 2\pi\kappa \frac{U_\infty}{L}$$

2. Nonuniform but steady vertical gust, relative to instantaneous location of the aircraft.

$$w = A \sin\left(\frac{2\pi(x+x_o-U_\infty t)}{\lambda}\right)$$

$$p = \frac{-2\pi AU_\infty \rho(z-z_o)}{\lambda} \cos\left(\frac{2\pi(x+x_o-U_\infty t)}{\lambda}\right)$$

3. Nonuniform but steady vertical gust, relative to the aircraft.

$$w = A \sin\left(\frac{2\pi(x+x_o)}{\lambda}\right)$$

$$p = \frac{-2\pi AU_\infty \rho(z-z_o)}{\lambda} \cos\left(\frac{2\pi(x+x_o)}{\lambda}\right)$$

In the gusts given above, κ is the reduced frequency and λ is the wavelength. A range of values for κ and λ will be used to try to determine the robustness of each boundary condition.

Test Cases

Once the ability to maintain a prescribed gust has been demonstrated a simple shear flow will be tested. In this problem, a two-dimensional airfoil is subjected to an incoming shear flow. The results of this study can be compared with the work of Zedan [17].

The next step will be to subject a simple body to the prescribed gusts. The 'Jupiter' nose cone [1] was chosen for these test since the geometry is currently being used for dynamical stability studies at AEDC. Furthermore, this geometry is simple to test in the ballistic range [3].

The first test with the 'Jupiter' nose cone will be a comparison of an unsteady gust (1) with the body fixed and a plunging body $\dot{z} = A \sin(\omega t)$ with steady far-field boundaries. These conditions have been set to produce the same angle of attack. This study will hopefully show effects which are not Galilean invariant.

The 'Jupiter' nose cone will then be used in a time variation study. The results of the unsteady gust will be compared to the body subject to the nonuniform but steady gust (2). The velocity of the body will result in the same time variation of the time varying wind case.

The final test case using the 'Jupiter' nose cone will involve the nonuniform but steady gust (3). This configuration will illustrate the effects of a body fixed gradient. This problem is similar to the case of a store in the flow field of an aircraft.

Concluding Remarks

In this paper, current boundary conditions were reviewed. These boundary conditions are typically based on one-dimensional characteristic theory. Extensions to allow for a nonuniform unsteady inflow boundary conditions were then proposed. Test cases were then presented which will demonstrate the validity of these conditions to correctly account for the specified inflow. Finally, some test cases were discussed which will allow for comparison with experimental work.

References

- [1] Bauer, R. C. and Riddle, C. D. "Transonic Dynamic Stability Tests of a Small Scale Model of the ABMA 'Jupiter' W14 Nose Cone Forebody in Combination with Three Afterbodies". AEDC-TN-58-42, July 1958.
- [2] Bayliss, A. and Turkel, E. "Far Field Boundary Conditions for Compressible Flows". *Journal of Computational Physics*, 48:182-199, November 1982.
- [3] Chapman, G. T. "The Ballistic Range as a Tool for Validation of Unsteady Computational Fluid Dynamic Codes". AFOSR/SFRP Final Report, July 1995.
- [4] Giles, M. B. "Nonreflecting Boundary Conditions For the Euler Equations". *AIAA Journal*, 28(12):2050-2058, December 1990.
- [5] Hirsch, C. *Numerical Computation of Internal and External Flows*, volume II. John Wiley and Sons, Inc., New York, NY, 1990.
- [6] Hirsch, C. *Numerical Computation of Internal and External Flows*, volume I. John Wiley and Sons, Inc., New York, NY, 1990.
- [7] Hoffman, K. A. and Chiang, S. T. *Computational Fluid Dynamics for Engineers*, volume II. Engineering Education System, Wichita, KS, 1993.
- [8] Jacocks, J. L. "An Investigation of the Aerodynamic Characteristics of Ventilated Test Section Walls for Transonic Wind Tunnels". PhD thesis, University of Tennessee, Knoxville, December 1976.
- [9] Pulliam, T. H. "Characteristic Boundary Conditions For the Euler Equations". NASA Publication 2201, Numerical Boundary Condition Procedures, October 1980.
- [10] Rodman, L. C. "The Application of Nonreflecting Boundary Conditions to 2-D Unsteady Computations on Curvilinear Grids". AIAA-90-1587, 1990.
- [11] Roe, P. L. "Remote Boundary Conditions for Unsteady Multidimensional Aerodynamic Computations". *Journal of Computers and Fluids*, 17(1):221-231, January 1987.
- [12] Saxer, A. P. and Giles, M. B. "Quasi-Three-Dimensional Nonreflecting Boundary Conditions For the Euler Equations". *Journal of Propulsion and Power*, 9(2):263-271, March-April 1993.
- [13] Thompson, K. W. "Time Dependent Boundary Conditions for Hyperbolic Systems". *Journal of Computational Physics*, 68(1):1-24, January 1987.
- [14] Thompson, K. W. "Time Dependent Boundary Conditions for Hyperbolic Systems, II". *Journal of Computational Physics*, 89(2):439-461, August 1990.
- [15] Yee, H. C. "Numerical Approximations of Boundary Conditions With Applications to Inviscid Equations of Gas Dynamics". NASA TM 81265, March 1981.
- [16] Yee, H. C., Beam, R. M., and Warming, R. F. "Stable Boundary Approximations For a Class of Implicit for the One-Dimensional Inviscid Equations of Gas Dynamics". AIAA-81-1009, March 1981.
- [17] Zedan, M. F. "A Method of Calculating Inviscid Flow around Arbitrary Airfoils in Upstream Shear". *Transactions of the Japan Society for Aeronautics and Space Sciences*, 38(119):1-15, May 1995.

Appendix

The flux Jacobian matrices, of equation (8), can be written as

$$\begin{bmatrix} \kappa_t & \kappa_x & \kappa_y & \kappa_z & 0 \\ \kappa_x \phi^2 - u\theta & \vartheta - \kappa_x(\gamma - 2)u & \kappa_y u - \kappa_x(\gamma - 1)v & \kappa_z u - \kappa_x(\gamma - 1)w & \kappa_x(\gamma - 1) \\ \kappa_y \phi^2 - v\theta & \kappa_x v - \kappa_y(\gamma - 1)u & \vartheta - \kappa_y(\gamma - 2)v & \kappa_z v - \kappa_y(\gamma - 1)w & \kappa_y(\gamma - 1) \\ \kappa_z \phi^2 - w\theta & \kappa_x w - \kappa_z(\gamma - 1)u & \kappa_y w - \kappa_z(\gamma - 1)v & \vartheta - \kappa_z(\gamma - 2)w & \kappa_z(\gamma - 1) \\ -\theta(\gamma e - 2\phi^2) & \kappa_x \psi - (\gamma - 1)u\theta & \kappa_y \psi - (\gamma - 1)v\theta & \kappa_z \psi - (\gamma - 1)w\theta & \kappa_t + \gamma\theta \end{bmatrix}$$

where $\theta = u\kappa_x + v\kappa_y + w\kappa_z$, $\phi^2 = \frac{(\gamma-1)}{2}(u^2 + v^2 + w^2)$, $\vartheta = \kappa_t + \theta$, and $\psi = (\gamma e - \phi^2)$. In the above equation, $\kappa = \xi$, η , or ζ for \hat{A} , \hat{B} , or \hat{C} respectively.

The eigensystem decomposition of the three-dimensional flux Jacobians have the form $\hat{A} = T_\xi \Lambda_\xi T_\xi^{-1}$, $\hat{B} = T_\eta \Lambda_\eta T_\eta^{-1}$, and $\hat{C} = T_\zeta \Lambda_\zeta T_\zeta^{-1}$. The matrix Λ_κ is the diagonal matrix whose elements are the eigenvalues of the flux Jacobian matrix. These eigenvalues are given by

$$\Lambda_\kappa = \begin{bmatrix} \vartheta & & & & \\ & \vartheta & & & \\ & & \vartheta & & \\ & & & \vartheta + a\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2} & \\ & & & & \vartheta - a\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2} \end{bmatrix}$$

where a is the speed of sound and $\vartheta = U, V$, or W for $\kappa = \xi, \eta$, or ζ respectively. The matrix T_κ , whose columns are the right eigenvectors of the flux Jacobians, is

$$\begin{bmatrix} \tilde{\kappa}_x & \tilde{\kappa}_y & \tilde{\kappa}_z & \alpha & \alpha \\ \tilde{\kappa}_x u & \tilde{\kappa}_y u - \tilde{\kappa}_z \rho & \tilde{\kappa}_z u + \tilde{\kappa}_y \rho & \alpha(u + \tilde{\kappa}_x a) & \alpha(u - \tilde{\kappa}_x a) \\ \tilde{\kappa}_x v + \tilde{\kappa}_z \rho & \tilde{\kappa}_y v & \tilde{\kappa}_z v - \tilde{\kappa}_x \rho & \alpha(v + \tilde{\kappa}_y a) & \alpha(v - \tilde{\kappa}_y a) \\ \tilde{\kappa}_x w - \tilde{\kappa}_y \rho & \tilde{\kappa}_y w + \tilde{\kappa}_z \rho & \tilde{\kappa}_z w & \alpha(w + \tilde{\kappa}_z a) & \alpha(w - \tilde{\kappa}_z a) \\ \frac{\tilde{\kappa}_x \phi^2}{\gamma-1} + \rho(\tilde{\kappa}_z v - \tilde{\kappa}_y w) & \frac{\tilde{\kappa}_y \phi^2}{\gamma-1} + \rho(\tilde{\kappa}_x w - \tilde{\kappa}_z u) & \frac{\tilde{\kappa}_z \phi^2}{\gamma-1} + \rho(\tilde{\kappa}_y u - \tilde{\kappa}_x v) & \alpha \left[\frac{\phi^2 + a^2}{\gamma-1} + \tilde{\theta} a \right] & \alpha \left[\frac{\phi^2 + a^2}{\gamma-1} - \tilde{\theta} a \right] \end{bmatrix}$$

where $\alpha = \frac{\rho}{\sqrt{2}a}$, $\tilde{\kappa}_x = \frac{\kappa_x}{\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}}$, $\tilde{\kappa}_y = \frac{\kappa_y}{\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}}$, $\tilde{\kappa}_z = \frac{\kappa_z}{\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}}$, and $\tilde{\theta} = \frac{\theta}{\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}}$.

The corresponding T_κ^{-1} , whose rows are the left eigenvectors of the flux Jacobians, is

$$\begin{bmatrix} \tilde{\kappa}_x \varphi - \frac{1}{\rho}(\tilde{\kappa}_z v - \tilde{\kappa}_y w) & \tilde{\kappa}_x(\gamma - 1)\frac{u}{a^2} & \tilde{\kappa}_x(\gamma - 1)\frac{v}{a^2} + \frac{\tilde{\kappa}_z}{\rho} & \tilde{\kappa}_x(\gamma - 1)\frac{w}{a^2} - \frac{\tilde{\kappa}_y}{\rho} & \tilde{\kappa}_x(\gamma - 1)\frac{1}{a^2} \\ \tilde{\kappa}_y \varphi - \frac{1}{\rho}(\tilde{\kappa}_x w - \tilde{\kappa}_z u) & \tilde{\kappa}_y(\gamma - 1)\frac{u}{a^2} - \frac{\tilde{\kappa}_z}{\rho} & \tilde{\kappa}_y(\gamma - 1)\frac{v}{a^2} & \tilde{\kappa}_y(\gamma - 1)\frac{w}{a^2} + \frac{\tilde{\kappa}_x}{\rho} & \tilde{\kappa}_y(\gamma - 1)\frac{1}{a^2} \\ \tilde{\kappa}_z \varphi - \frac{1}{\rho}(\tilde{\kappa}_y u - \tilde{\kappa}_x v) & \tilde{\kappa}_z(\gamma - 1)\frac{u}{a^2} + \frac{\tilde{\kappa}_y}{\rho} & \tilde{\kappa}_z(\gamma - 1)\frac{v}{a^2} - \frac{\tilde{\kappa}_x}{\rho} & \tilde{\kappa}_z(\gamma - 1)\frac{w}{a^2} & \tilde{\kappa}_z(\gamma - 1)\frac{1}{a^2} \\ \beta(\phi^2 - \tilde{\theta}a) & -\beta[(\gamma - 1)u - \tilde{\kappa}_x a] & -\beta[(\gamma - 1)v - \tilde{\kappa}_y a] & -\beta[(\gamma - 1)w - \tilde{\kappa}_z a] & \beta(\gamma - 1) \\ \beta(\phi^2 + \tilde{\theta}a) & -\beta[(\gamma - 1)u + \tilde{\kappa}_x a] & -\beta[(\gamma - 1)v + \tilde{\kappa}_y a] & -\beta[(\gamma - 1)w + \tilde{\kappa}_z a] & \beta(\gamma - 1) \end{bmatrix}$$

where $\beta = \frac{1}{\sqrt{2}\rho a}$ and $\varphi = \left(1 - \frac{\phi^2}{a^2}\right)$.

A FLUORESCENCE BASED
FIBER-OPTIC OIL VISCOSITY MONITOR

Christopher M. Stellman
Ph.D. Candidate
Department of Chemistry and Biochemistry

University of South Carolina
Physical Science Center
Columbia, SC 29208

Final Report for
Graduate Research Program
Arnold Engineering Development Center

Sponsored by
Air Force Office of Scientific Research
Bolling Air Force Base, Washington D.C.
and
Arnold Engineering Development Center
Arnold Air Force Base, Tullahoma, TN

August 1995

A FLUORESCENCE BASED
FIBER-OPTIC OIL VISCOSITY MONITOR

Christopher M. Stellman
Ph.D. Candidate
Department of Chemistry and Biochemistry
University of South Carolina

Abstract

A fiber-optic spectroscopy system for the on-line determination of oil viscosity in industrial machinery is proposed. The instrument monitors spectral changes in oil fluorescence as a function of viscosity. Multivariate statistics provide an ideal method of tracking this phenomenon. A detailed description of the instrument's design is presented and a foundation for its calibration is laid. Finally, a preliminary experiment to evaluate the instrument's performance is proposed.

Introduction

Lubricant oil quality is essential in preserving the performance and longevity of industrial machinery. Viscosity is a good indicator of oil condition and performance.^{1,2} However, conventional methods of oil viscosity determination are time and labor intensive. This has resulted in the practice of replacing oil at regular intervals (regardless of its condition) or periodic viscosity testing (often with unacceptable frequency). These practices can lead to economic losses through the unnecessary replacement of good quality oil or more seriously through the damage of mechanical components resulting from undetected oil degradation. In an industrial environment on-line oil viscosity determination may prove crucial to improving lifetime engine performance in a cost-labor-and-time efficient manner. Fiber-optic fluorescence spectroscopy is uniquely applicable to such a monitor.

For maximum effectiveness the monitor must be capable of producing a measure of oil viscosity in near-real time. It is desirable that the system's hardware be mechanically simple, robust, portable and inexpensive. The instrument should also have the ability to monitor viscosity remotely (via a small probe) to insure facile system interfacing in the often cramped and hostile environment of industrial machinery.

The following document proposes a fiber-optic spectroscopy system for the on-line determination of oil viscosity. The instrument monitors spectral changes in oil fluorescence as a function of viscosity. Multivariate statistics provide an ideal method of tracking this phenomenon. A detailed description of the instrument's design is presented and a foundation for its calibration is laid. Finally, a preliminary experiment to evaluate the instrument's performance is proposed.

Background

Conventional capillary-flow, falling-ball and rotating-disk viscometers require large sample volumes and are difficult to implement for on-line viscosity determination. The systems must be carefully designed into the framework of machinery, making them difficult to incorporate into existing mechanical systems. Furthermore, the instruments are not conducive to on-line monitoring of remote and/or spatially

confined locations.³ Vibrational and ultrasonic viscometers have improved upon these hurdles but are still limited in their ability to be readily interfaced with existing machinery.^{4,5}

Fiber-optic spectroscopy has been used to remotely monitor numerous chemical and physical phenomena.⁶⁻⁹ Small fiber-optic probes are resistant to severe environments (e.g. magnetic fields, high temperatures, etc.) and provide an ideal interface between the analyte of interest and spectroscopic hardware. Fiber-optics are flexible and can transmit light over long distances (e.g. 100 m) with little attenuation. This can isolate sensitive electronics (sensors) from the point of detection and allow for remote monitoring of harsh and/or spatially confined locations.

In recent years vibrational spectroscopy has been used to investigate numerous quality control parameters of petroleum-based products. The most useful techniques are mid-IR absorption, Raman scattering and UV fluorescence. Mid-IR absorption has been used to monitor the presence of water contaminants, the formation of oxidation products, and changes in total base number and viscosity.¹⁰⁻¹⁷ Raman scattering has proven useful in determining percent fuel composition, cetane number, and the general hydrocarbon makeup, aromatic components and additives of several gas-fuel products.¹⁸⁻²³ UV fluorescence has been used to confirm the presence of crude oil deposits, monitor tramp oil contaminants, and determine lubricant viscosity.²⁴⁻³⁰

All of the aforementioned techniques offer a viable alternative for on-line monitoring of oil viscosity. However, a number of practical advantages and disadvantages must be considered for each (Table 1). While all of the methods excel in specific areas, fluorescence spectroscopy offers a balance of performance best suited to an on-line oil viscosity monitor. Strong emphasis is given to the use of an evasive probe and the relatively simple, robust and inexpensive hardware associated with this technique.

Fluorescence

While a comprehensive theory of fluorescence is beyond the scope of this document, a brief insight into the viscosity-dependent fluorescence of oils is useful. It has been documented that fluorescence spectra of petroleum-based lubricants undergo a red-shift in emission wavelength with increasing viscosity.^{24,25}

As an oil wears and viscosity increases, oxidative degradation of the oil's macromolecular chain increases solvent polarity and changes the oil's optical constants. In particular, the dielectric constant is increased and induces a shift in emission to lower energy (red-shift). It is on this phenomenon that fluorescence based oil viscosity monitoring is dependent. Equation 1 shows the relationship between average emission energy (E_e), excitation energy (first term on the right), the effects of permanent dipole interactions (second term on the right) and the effects of induced dipole interactions (third term on the right).

$$(1) \quad E_e = E - \left[\left(\frac{2}{a^3} \right) \left(\frac{D-1}{D+2} - \frac{n^2-1}{n^2+2} \right) (\mu_e (\mu_e - \mu_o \cos \alpha)) \right] - \left[\left(\frac{1}{a^3} \right) \left(\frac{n^2-1}{n^2+2} \right) (\mu_e^2 - \mu_o^2) \right]$$

More specifically Equation 1 shows that emission energy is dependent on the excitation energy (E), the sample volume (a^3), the static dielectric constant (D), the refractive index (n), the excited state dipolar moment (μ_e), the ground state dipolar moment (μ_o) and the angle between the dipolar moments (α). It is evident from Equation 1 that as the oil's dielectric constant increases (viscosity increases) there is a corresponding decrease in emission energy (a red-shift).

Appropriate excitation and emission wavelengths are critical to optimizing the performance of a fluorescence based analysis technique. In terms of radiant energy, maximum emission from minimum excitation is desired. However, consideration should also be given to the amount of spectral variance (more is better) over a given sample viscosity range. It has been documented that oil emission spectra generally show an increase in viscosity-dependent red-shifting with decreasing excitation wavelength.²⁵ Optimum wavelength parameters for a given sample can be determined via excitation-emission spectroscopy.³¹ An excitation spectrum is a plot of fluorescence signal versus excitation wavelength (with a constant emission wavelength and bandpass) and an emission spectrum is a plot of fluorescence signal versus emission wavelength (with a constant excitation wavelength and bandpass). The two spectra are used to determine the best excitation and emission wavelengths respectively. A more complete picture of a samples spectral characteristics can be obtained with total fluorescence spectroscopy.³¹ This technique

provides a three-dimensional plot of the fluorescence signal as a function of both the excitation and emission wavelengths. Like excitation-emission spectroscopy, total fluorescence is an established technique for determining optimum wavelength parameters. Instrumentation for both techniques are prevalent in modern spectroscopic laboratories, making this one-time wavelength optimization a practical endeavor.

Changes in the viscosity of polyphenyl ether lubricants has previously been monitored via fluorescence spectroscopy with an excitation and emission wavelength of 340 nm and 362 nm respectively.²⁴ The following instrumental design is based on (but not limited to) these wavelengths.

Instrumental Design

The instrument (Figure 1) consists of a UV excitation source, a two fiber probe and a multichannel spectrofluorometer. UV radiation is filtered with a bandpass filter of desired wavelength and launched into the proximal end of the probes excitation fiber. At the probe tip, induced fluorescence is collected via the distal end of a collection fiber and transmitted to the spectrofluorometer. The light is filtered with a longpass filter to remove Rayleigh scattering, dispersed with a ruled grating and focused onto a charge-coupled device (CCD) detector. The systems experimental parameters and data analysis are controlled via a portable personal computer. The following gives a detailed explanation of the instrumental components, closing with a brief note on instrumental cost.

Excitation. A 75 W Xe arc lamp (Oriel, Q Series) provides an intense spectral continuum over the UV, visible and near-IR wavelength regions (200-1000 nm). The use of a continuum source allows for versatility in excitation wavelength. Any wavelength over the lamps spectral distribution may be selected with the appropriate choice of bandpass filter. The 6x6x9 inch aluminum lamp housing is both compact and rugged. The lamp is operated by a dc power supply and employs internal circuitry to provide for light ripple of less than 0.8% rms.

A bandpass filter provides excitation at 340 nm with a minimum transmission of 20% and a spectral bandpass of 20 nm. The filter's good transmission and compatibility with fast collection optics

makes for a high-throughput wavelength selector. The filter is mounted prior to the focusing optics and is easily replaced with any two inch filter of desired wavelength and bandpass.

After filtering, radiation is focused onto the proximal end of the excitation fiber with a fused silica condensing lens. Fused silica provides >90% transmission from 200-1000 nm and is resistive to the high temperatures associated with arc sources. The lens' F# (1.8) closely matches that of the fiber (2.2) to provide a high collection efficiency.

UV-enhanced silica fibers (400 μm -core) serve as an optical link between the excitation source, sampling probe and spectrofluorometer. The high-OH fibers have 98% transmission per meter at 350 nm and are substantially less expensive than their low-OH counterparts. The fibers are encased in a polyolefin tubing to prevent fiber breakage and better facilitate handling. The proximal end of the excitation fiber is fitted with an SMA connector and mounted to the lamp via a fiber-optic input assembly. The distal end of the collection fiber is directly coupled to the sampling probe.

Sampling. The sampling probe's design is shown in Figure 2 (top). The optical housing consists of the distal ends of the excitation and collection fibers sealed in a stainless steel tube (800 μm i.d.) with five-minute epoxy. The housing is centered in a bored-out 1/8 inch hole running lengthwise through a 1/4 inch hex bolt. The hole is sealed with five-minute epoxy and the probe's face is polished smooth. The bolt has a standard thread (20 per inch) to provide facile system interfacing with existing machine ports. An adapter may be used to access holes of larger diameter and/or different threads.

Fluorescence spectroscopy allows for the collection of light with a backscattering (180 °) geometry, making the use of an evasive probe acceptable. This is not true of transmission-based spectroscopies such as mid-IR absorption. The design differences between an evasive and invasive probe (Figure 2) help to demonstrate the advantages and disadvantages of each respectively. The evasive probe consists of a simple two fiber design while the invasive probe employs a series of fibers, windows, lenses and mirrors. As a result the evasive probe is less sensitive to mechanical shock and/or changing temperature and pressure. The evasive probe also lacks an optical cavity making it immune to particulate

clogging. Finally, its non-intrusive design is conducive to monitoring spatially confined regions (e.g. gear boxes, oil transfer lines, etc.) without disturbing lubricant flow.

Detection. The fluorescence signal is transmitted to the spectrofluorometer via the collection fiber. The proximal end of the fiber is fitted with an SMA connector and directly coupled to an 1/8 m imaging spectrograph (Oriel, MS125) via a fiber-optic F# matching adapter. The adapter uses a spherical lens to "match" the F# of the fiber (2.2) to that of the spectrograph (3.8), optimizing collection efficiency and increasing instrument throughput. Rayleigh line attenuation is achieved by placing a 350 nm longpass filter at the proximal end of the fiber. The filter provides >90% transmission above 350 nm and like the bandpass filter it is easily replaced with any one inch filter of desired optical properties.

The 6x6x3 inch spectrograph provides a compact and robust method of wavelength dispersion. A 600 line/mm ruled grating is blazed at 400 nm, has an effective wavelength region of 250-1300 nm, and a peak efficiency of 85%. Gratings of other spectral, bandpass and resolution requirements are available. A precision holder, which clips into a grating mount, supports the grating and allows it to be easily interchanged. The grating is translated with a hand driven micrometer, making for a rugged yet practical method of spectral alignment.

The fluorescence spectrum is dispersed onto the image plane of the CCD detector. CCDs are known for their high quantum efficiency, imaging capability, long lifetimes, low noise and solid-state characteristics. Historically, the disadvantage of spectroscopy-specific CCDs has been their relatively high-cost. However, it has recently been demonstrated that a low-cost CCD (originally designed for astronomy) can be used for spectroscopic measurements (e.g. Raman, fluorescence) with only slight modification to the detector's cooling system.^{18,32} Upon modification, the CCD provides a more than adequate level of detection performance at 20-30% the cost of conventional spectroscopy-specific detectors.

The CCD (Santa Barbara Instrument Group, ST-6V) provides enhanced UV spectral response out to 200 nm and has a resolution of 750 x 242 (H x V) elements with element sizes of 11.5 x 27 μm . The ST-6V employs a two-stage thermoelectric cooler to chill the detector, with an air cooled anodized heat sink. Operating in ambient conditions near 26° C the minimum sustainable detector temperature is near

-20° C. To obtain lower temperatures a 4 foot section of 1/4 inch copper tubing is wrapped around the CCD housing and held in place with two 8 inch screw clamps. Rubber hoses are coupled to each end of the tubing, placed in an ice bath and water is circulated with an aquarium-style pump. For long term cooling the ice bath can be replaced with a commercial heat exchanger/circulator. The modification provides for minimum temperatures of -50° C to be easily obtained at 80% of the maximum TE cooler current. The increased cooling capacity allows for rapid collection (e.g. 10-30 second exposures) of good quality spectra by reducing the dark count noise (thermally induced charge) of the detector. Cements used in the ST-6V detector head to connect the CCD chip to the TE cooler and the TE cooler to the heat sink are not rated below -50°C. Therefore, operation below this temperature is not recommended.

The instrument is controlled with a portable personal computer. The operating system provides a means of performing instrument calibration, controlling experimental parameters and analyzing collected data. Component interfacing needs and operating system requirements are better realized through discussion of the proposed data analysis and calibration method (see below).

Cost. The proposed instrument has a base cost of \$16,000. A cost breakdown is given in Table 2. No projection is given for labor costs and/or expenses attributed to expendable materials (e.g. epoxy sealants, oil samples, etc.). The author has recommended components based on experience with manufacturer quality and system compatibility. Comparable equipment is often available from alternative sources, possibly at a lower cost. Also, many components can be fabricated by the developer (e.g. fiber-optic mounts, condensing/collimating systems, filter holders, etc.) depending on manufacturing capabilities.

Data Analysis

Traditional fluorescence based analysis is univariate in nature, meaning that the intensity of emission at a given wavelength (usually an emission maximum) is used to monitor an analyte of interest. A univariate analysis focuses on the variance of a single channel (wavelength) and does not take advantage of a spectrum's inherent multivariate profile. In short, potentially useful data is thrown away. Variability in a spectral data set is best analyzed by a multivariate method such as Principal Component Analysis (PCA).

PCA attempts to find a series of abstract spectra called principal components (PCs) which may be weighted in different proportions and summed to reproduce the original spectral data. Each spectrum can thus be represented by a set of principal component weightings called scores resulting in considerable data reduction.³³

In brief, spectral data of n channels per spectrum can be viewed as a point in an n -dimensional space. PCA performs a coordinate transformation which expresses the variability of a data set in the fewest possible dimensions. Accounting for all the spectral variability requires a number of PCs equal to the smaller of the number of channels or the number of samples.^{33,34} The amount of explained variability decreases for each additional PC; beyond the number of PCs necessary to explain real variability, the remaining PCs consist mostly of "noise". The first PC (PC1) is the least-squares line that best fits the variability of the data. The second (PC2) and higher PCs are orthogonal to PC1 and are ranked in decreasing order of correlation with the data variability. PCA decomposition of the data matrix can be obtained by eigenvalue-eigenvector analysis. The resulting eigenvalues are measures of the variance explained by each PC. After the information in a data matrix is compressed to its most relevant features by PCA, the sample spectra can be regressed on (fitted to) physical properties (e.g. viscosity) for calibration purposes. This process is termed principal component regression (PCR).^{33,34} PCA and PCR of collected spectra are easily performed using a commercially available statistics software package such as "Pirouette" (Infometrix).

Calibration

Determination of oil viscosity via fluorescence spectroscopy requires that the unknown sample's spectrum be referenced to a calibration set. Calibration consists of several steps which are outlined in Figure 3. First, a spectrum is recorded for each sample in the calibration set. A typical calibration set consists of 15-25 samples covering a known range of viscosity (see below for sample preparation). PCA of the spectra provides a number of PCs that describe the spectral variance over the given viscosity range. Analysis of multiple PC's is readily achieved using the varimax method (rotation of PCs) but is difficult to

visualize in two-dimensional space. Therefore, for simplicity of discussion it is assumed that the loading of PC1 describes a large majority (>90%) of the spectral variance. This is often the case. Next, each spectrum is scored against the loading of PC1. This is essentially a weighting of the spectrum's similarity to that of the loading plot. Finally, by projecting the spectra's scores against their corresponding viscosities a plot of PC score as a function of viscosity is obtained. This relationship is used to predict the viscosity of unknown samples, provided that their viscosity lies within the bounds defined by the calibration set.

Experimental

Prior to use as an on-line monitor it is essential that the instrument's performance is well characterized. Two important performance considerations of any spectroscopic based monitoring system are the precision and accuracy to which the analytical method can quantitate the measure of interest (e.g. viscosity). For the proposed system, precision and accuracy are best determined through statistical comparison of the actual and predicted viscosities for both a sample calibration set and an independent validation set.

A calibration and validation set are best prepared with federal test method 5307.2.³⁵ In brief, the oil is subjected to a specified oxidative, thermal and catalytic environment for a standard test duration of 96 hours. Exposure time can be increased or decreased for larger or smaller viscosity ranges respectively. Test temperatures are dependent on the type of lubricant and the desired rate of viscosity breakdown. At periodic intervals (depending on the number of samples desired) 10 mL samples are extracted. Kinematic viscosity is then determined with ASTM test method 445.³⁶ The test measures the time required for a fixed volume of liquid (oil) to flow under gravity through the capillary of a calibrated viscometer. The test is performed under a reproducible driving head and known temperature. If desired, dynamic viscosity can be determined by multiplying the measured kinematic viscosity by the density of the lubricant.

The first data set is used to calibrate the instrument and the second set is used to validate whether the instrument can predict the viscosity of unknown samples. It should be noted that the validation test set

must consist of unique samples. It should not be created by simply dropping samples from the original calibration set.

Spectra are collected for each of the prepared sample sets and subjected to appropriate multivariate analysis (see above). By plotting the actual versus the predicted viscosities for each data set a measure of "goodness of fit" can be determined. For the calibration set this measure is determined by calculating the correlation coefficient and standard error of estimate (SEE) and for the validation set a measure of standard error of prediction (SEP) is required.

The correlation coefficient gives a measure of the closeness to linearity of a relationship between two variables (e.g. actual and predicted viscosity) and determines the precision with which the determined regression line can be used for prediction.³⁷ The correlation coefficient (r^2) can be determined from Equation 2:

$$(2) \quad r^2 = \frac{\sum (Y_p - Y_m)^2}{\sum (Y_a - Y_m)^2}$$

where Y_p is the predicted viscosity, Y_a is the actual viscosity and Y_m is the mean viscosity.

The SEE and SEP are measures of inaccuracy resulting from "experimental errors", consisting of a combination of errors due to factors not controlled by the experiment, errors in measurement and possibly round-off errors.³⁷ The distinction between SEE and SEP is that they are determined for the calibration and independent validation sets respectively. The standard error (s_e) for each can be determined from Equation 3:

$$(3) \quad s_e = \sqrt{\frac{\sum (Y_a - Y_p)^2}{n - 2}}$$

where Y_p is the predicted viscosity, Y_a is the actual viscosity and n is the total number of samples.

Summary

A fiber-optic spectroscopy system for the on-line determination of oil viscosity in industrial machinery has been proposed. The instrument monitors spectral changes in oil fluorescence as a function of viscosity. Multivariate statistics provide an ideal method of tracking this phenomenon. A detailed description of the instrument's design has been presented. An emphasis has been placed on mechanical simplicity, robustness, portability and cost. Fiber-optics have been employed to provide remote capabilities. A foundation has been set for using PCA and PCR of spectral data to calibrate the instrument and provide subsequent analysis of unknowns. Finally, a preliminary experiment to evaluate the instrument's performance has been proposed. The experiment statistically measures the precision and accuracy with which the instrument can quantitate viscosity.

<u>TABLE 1</u>	MID-IR ABSORPTION	RAMAN SCATTERING	UV-FLUORESCENCE
SPECTRAL INFORMATION AND DOMINANT USES IN SAMPLE ANALYSIS	(1) Good fingerprint region provides a wealth of structural information. (2) Primarily used for qualitative identification and molecular structure determination. (3) Used in quantitative mixture analysis and reaction studies.	(1) Provides a wealth of structural information. (2) Primarily used for qualitative identification and molecular structure determination. (3) Used in quantitative determinations of minor trace constituents.	(1) Broad peaks provide minimal structural information. (2) Primarily used for the study of chemical equilibria and kinetics. (3) Less frequently used in qualitative analysis.
USE OF <i>IN-SITU</i> FIBER-OPTIC PROBE	(1) Possible. (2) Probe must be invasive. (3) Design employs many optical components. (4) Probe is less robust in design.	(1) Possible. (2) Probe can be evasive. (3) Design employs few optical components. (4) Probe is more robust in design.	(1) Possible. (2) Probe can be evasive. (3) Design employs few optical components. (4) Probe is more robust in design.
INSTRUMENT SIMPLICITY	(1) Mechanically complex. (2) Cost prohibits using multichannel detectors. (3) Use of choppers and interferometers make for numerous moving parts.	(1) Mechanically simple. (2) Multichannel detector preferred for rapid on-line analysis. (3) Minimal to no Moving parts.	(1) Mechanically simple. (2) Multichannel detector preferred for rapid on-line analysis. (3) Minimal to no moving parts.
INSTRUMENT PORTABILITY AND ROBUSTNESS	(1) Limited portability. (2) Moving parts make for a less robust instrument.	(1) Good portability. (2) Lack of moving parts makes for robust instrument.	(1) Good portability. (2) Lack of moving parts makes for robust instrument.
COST	(1) Moderate to high.	(1) Moderate to high.	(1) Moderate.

FIGURE 1

Instrumental design of a fluorescence-based fiber-optic oil viscosity monitor:
(a) arc lamp (b) lens (c) bandpass filter (d) fiber mount (e) fibers (f) probe (g)
edge filter (h) mirror (i) grating and (j) detector (k) computer (l) power supply.

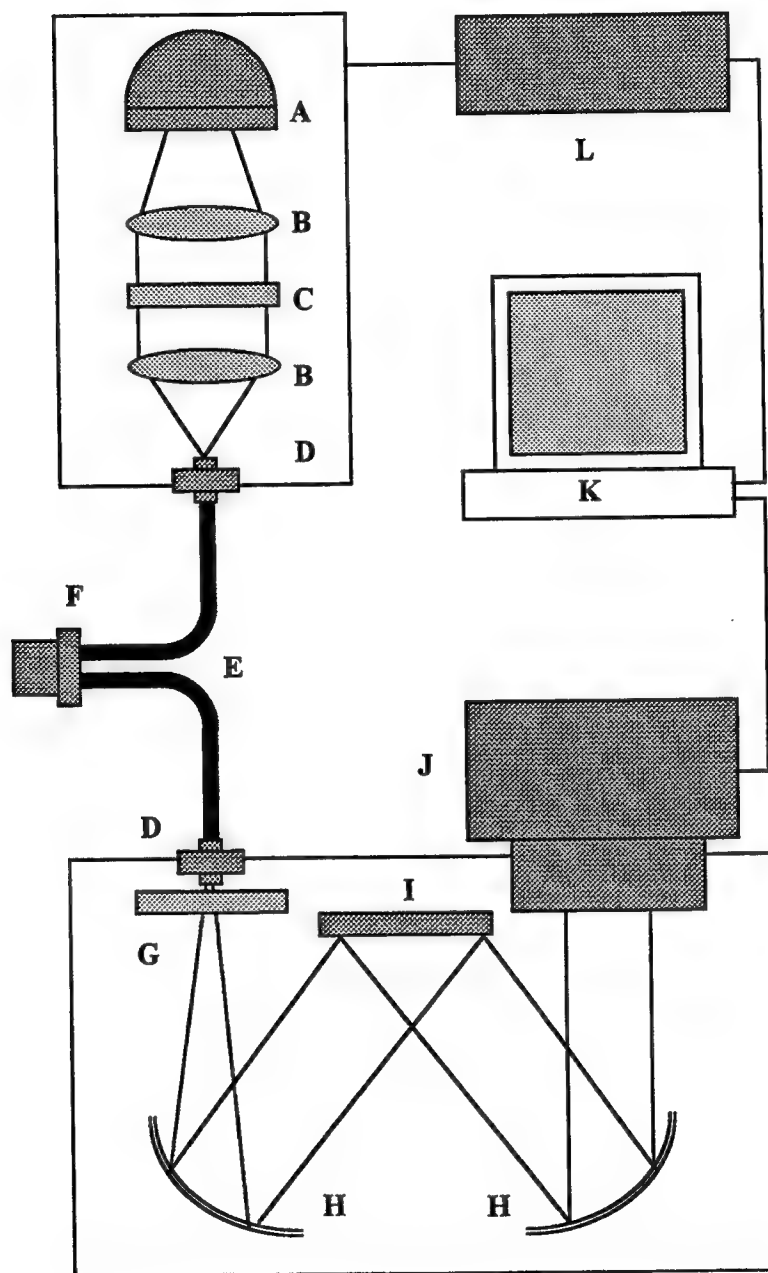
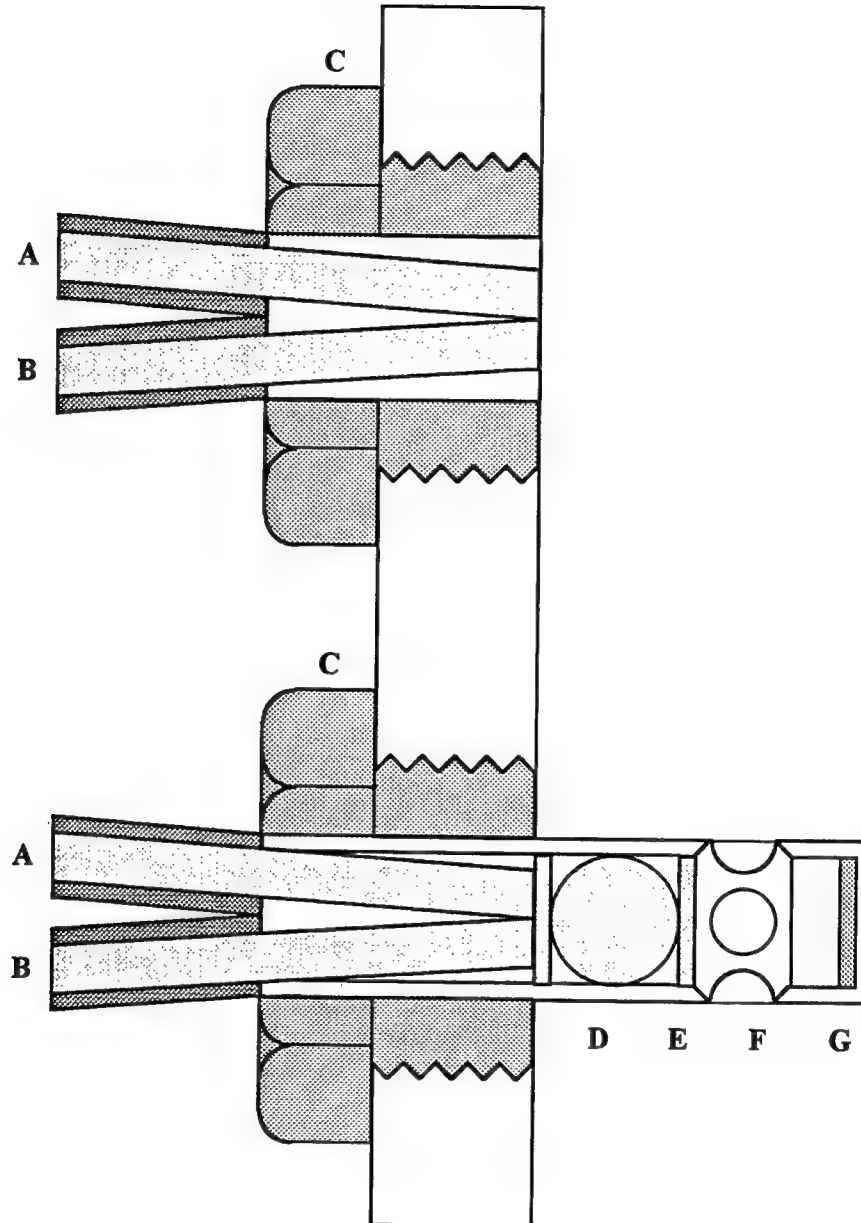


FIGURE 2

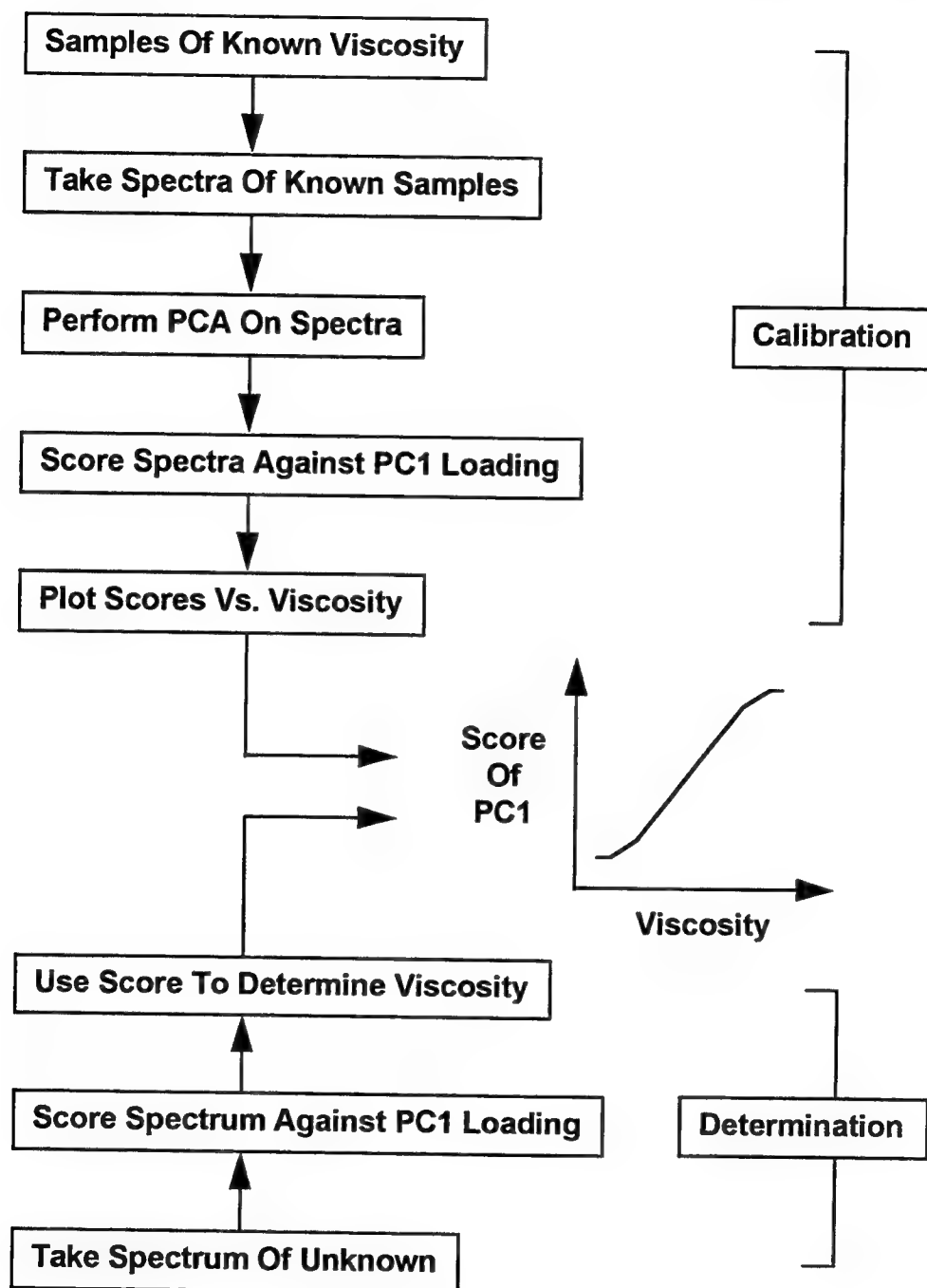
Cross section of an evasive (top) and invasive (bottom) probe: (a) source fiber (b) collection fiber (c) bored out hex nut (d) glass sphere (e) glass window (f) sample inlet holes (g) rear silvered mirror.



<u>TABLE 2</u>	MANUFACTURER	MODEL	COST (US\$)
SOURCE COMPONENTS			
75 W Xe Arc Lamp	Oriel Corporation	6263	200.00
Q series Lamp Housing	Oriel Corporation	60000	490.00
Power Supply	Oriel Corporation	68806	1430.00
Condenser/Collimator	Oriel Corporation	60007	320.00
Filter Holder	Oriel Corporation	62020	70.00
Filter Spacing Tube	Oriel Corporation	7129	60.00
Bandpass Filter	Oriel Corporation	58630	360.00
SAMPLING COMPONENTS			
Fiber Optics (100 m)	Polymicro Technologies	FVP500550590	500.00
SMA Connectors (10)	Thorlabs, Inc.	10410A	70.00
DETECTION COMPONENTS			
Edge Filter	Oriel Corporation	51260	40.00
Spectrograph (1/8 m)	Oriel Corporation	77400	1,600.00
Grating (400 nm Blaze)	Oriel Corporation	77414	260.00
Fiber F# Matcher	Oriel Corporation	77529	1,300.00
CCD Detector	Santa Barbara Instrument Group	ST-6V	4,800.00
OTHER			
Portable Personal Computer	N/A	N/A	2,000.00
Multivariate Software Package	Infometrix	Pirouette	2,500.00
TOTAL			16,000.00

FIGURE 3

Block diagram of steps performed during system calibration and steps performed during viscosity determination of an unknown sample.



References

- 1) Bureau of Naval Personnel, *Fundamentals of Petroleum* (U.S. Government Printing Office, Washington D.C., 1957).
- 2) K. Boldt and B.R. Hall, *Significance of Tests for Petroleum Products* (American Society for Testing and Materials, Philadelphia, 1984).
- 3) J.M. Dealy, *Measurement and Science*, October 1, 61 (1984).
- 4) S.H. Sheen, H.T. Chien and A.C. Raptis, U.S. Government Document, Argonne National Laboratory, Conf. 9407123-2 (1994).
- 5) R.W. Cernosek, S.J. Martin, K.O. Wessendorf, M.D. Terry and A.N. Rumpf, U.S. Government Document, Sandia National Laboratories, Conf. 9409202-1 (1994).
- 6) R.E. Dessy, *Anal. Chem.* 61, 1079A (1989).
- 7) R.W. Murray, R.E. Dessey, W.R. Heineman, J. Janata and W.R. Seitz, *Chemical Sensors and Microinstrumentation* (American Chemical Society, Washington D.C., 1989).
- 8) W.R. Seitz, *Crit. Revs. in Anal. Chem.* 19, 135 (1988).
- 9) S.M. Angel, *Spectroscopy* 2, 38 (1987).
- 10) M. Wiseman and A. Ah-Sue, *Lubr Eng.* March, 236 (1992).
- 11) A.D. Stuart, S.M. Trotman, K.J. Doolan and P.M. Fredericks, *Appl. Spectrosc.* 43, 55 (1989).
- 12) J.P. O'Hara, A.B. Sarkis and W.A. Kennedy, *Soc. Automot. Eng.*, Publ. No. 730745 (1973).
- 13) J.P. Coates and L.C. Setti, *Soc. Automot. Eng.*, SP-558, 37 (1983).
- 14) J.P. Coates, L.C. Setti and B.B. MCCaa, *Soc. Automot. Eng.*, SP-589, 81 (1984).
- 15) L. Gardner, *Canad. Spectrosc.* 11,98 (1966).
- 16) G.O. Dotterer and W.W. Hellmuth, *Lubr. Eng.*, 41, 89 (1985).
- 17) D.A. Compton, J.R. Young, R.G. Kollar, J.R. Mooney and J.G. Grasselli, *Computerized Quantitative Infrared Spectroscopy* (American Society for Testing and Materials, Philadelphia, 1984).
- 18) J.B. Cooper, P.E. Flecher, T.M. Vess and W.T. Welch, *Appl. Spectrosc.* 49, 586 (1995).
- 19) T. Shope, T. Vickers and C. Mann, *Appl. Spectrosc.* 42, 468 (1988).

- 20) M.B. Seasholtz, C.C. Archibald, A. Lorber and B.R. Kowalski, *Appl. Spectrosc.* 42, 1067 (1989).
- 21) K.P.J. Williams, R.E. Aries, D.J. Cutler and D.P. Lidiard, *Anal. Chem.* 62,2553 (1990).
- 22) W.M. Chung, Q. Wang, U. Sezerman and R.H. Clarke, *Appl. Spectrosc.* 45, 1527 (1990).
- 23) R.H. Clarke, W.M. Chung, Q. Wang, S. DeJesus and U. Sezerman, *J. Raman Spectrosc.* 22,79 (1991).
- 24) M.A. Keller and C.S. Saba, *Appl. Spectrosc.* 22, 266 (1990).
- 25) M.H. Vinas, E.M. Murciano, T. Montoro and J.L. Escudero, *Fiber Optic Medical and Fluorescent Sensors and Applications*, SPIE 1648, 238 (1992).
- 26) A.M. Williams and D. Ben-Amotz, *Anal. Chem.* 64, 700 (1992).
- 27) A.D. Thruston and R.W. Knight, *Environ. Science and Technol.* 5,64 (1971).
- 28) K.R. Januszkiewicz, R. Hombek and H.H. Sulek, *Lubr. Eng.* June, 448 (1991).
- 29) T.D. Downare and O.C. Mullins, *Appl. Spectrosc.* 49, 754 (1995).
- 30) O.C. Mullins, S. Mitra-Kirtley and Y. Zhu, *Appl. Spectrosc.* 46, 1405 (1992).
- 31) J.D. Ingle and S.R. Crouch, *Spectrochemical Analysis* (Prentice Hall, Englewood Cliffs, 1988).
- 32) J.B. Cooper, J. Aust, C. Stellman, K. Chike and M.L. Myrick, *Spectrochim. Acta*, 50A, 567 (1994).
- 33) C. Chatfield and A.J. Collins, *Introduction to Multivariate Analysis* (Chapman and Hall, New York, 1980).
- 34) I.T. Jolliffe, *Principal Component Analysis* (Springer-Verlag, New York, 1986).
- 35) U.S. government test Method, Corrosiveness and Oxidation Stability of Aircraft Turbine Engine Lubricants, Method 5307.2, Std. No. 791C, (1986).
- 36) ASTM test method, Standard Method for Kinematic Viscosity of Transparent and Opaque Liquids, Designation D 455-94, (1995).
- 37) N. Gilbert, *Statistics* (W.B. Saunders Company, Philadelphia, 1976).

UPDATES AND CHANGES TO
THE SIGNATURE MODEL FOR
INFRARED FLARES (SMIRF)

John R. Tomlinson
PhD Student
Department of Mathematics

Texas Tech University
Lubbock, TX 79409

Final Report for:
Graduate Student Research Program
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Arnold Engineering Development Center

August 1995

UPDATES AND CHANGES TO
THE SIGNATURE MODEL FOR
INFRARED FLARES (SMIRF)

John R. Tomlinson
PhD Student
Department of Mathematics
Texas Tech University

Abstract

The Signature Model for Infrared Flares (SMIRF) began construction in 1989 at the request of the then Air Force Electronic Warfare Center. In the intermediate years, SMIRF has received little attention from the Air Force or other sponsors. However, good work has been performed in the interim by researchers working under the auspices of various AFOSR summer programs. The goal of the research this summer was to integrate several new capabilities into SMIRF and to determine its suitability for further research into automatic error analysis within models. Results are mixed. One of the shortcomings identified during the summer's research was the severe lack of coherent documentation about the construction of SMIRF. While this researcher has been involved with SMIRF since before its inception, the task of reconstructing events and design background proved daunting. The primary outputs from this research, then, are several new capabilities in SMIRF along with new more complete documentation. Finally, SMIRF is deemed acceptable as a vehicle for further research into automatic generation of error analysis.

UPDATES AND CHANGES TO THE SIGNATURE MODEL FOR INFRARED FLARES (SMIRF)

John R. Tomlinson

Introduction

This report describes improvements to the Signature Model for Infrared Flares (SMIRF) computer program. The purpose of this research was to determine the possibility of using SMIRF as a basis for development of error propagation methodology. The methodology would not be limited to a single purpose model, such as a linear regression curve fit, but rather would try to encompass such model structures as differential equation solvers and finite element methods.

This report includes a discussion of the overall research objectives and how SMIRF fits into the research. Improvements to SMIRF resulting from the summer's research will be detailed. Some theoretical analysis of error within SMIRF's numerical techniques will be discussed and will justify some of the changes made to SMIRF as well as suggest further directions for the overall research objective. Finally, the current status of SMIRF will be described along with suggestions for future research.

Background

The primary purpose of this research is to develop a range of automatic error analysis techniques for computer models. Considerable progress has been made in this area with respect to simple models. For instance, if a linear regression fit is made from experimental data, the statistical distribution of the data can be used to predict the accuracy of the model in interpolating data points or even in extrapolating to new regions.¹ The analysis

¹For further information see Kendall Brown's report elsewhere in this document [1].

of stochastic processes falls within this arena. In another area, a linear model can propagate a known input error distribution to the output. These two cases illustrate different aspects to the problem of automatic error analysis within models. In the first case, the error analysis is performed to determine the "goodness-of-fit" of the model to the data. In the second, the error analysis provides insight into the effect input data variability causes in output results.

However, these models are fairly simple constructions. In the case of non-linear models or models which involve complicated differential equation solution techniques, the usual approach to determine the output variability is the use of a "monte carlo" technique.

The goal of this research is to determine numerical techniques which allow computer programs, without "monte carlo", to determine the output effect caused by input variability.

In 1989, the researcher asked Arnold Engineering Development Center to begin construction of a computer program which would model the signature and flight path of a burning flare ejected from an aircraft. The result was the Signature Model for Infrared Flares (SMIRF). SMIRF is a computer model which contains both complex and simple parts. The researcher hopes that the structure of the SMIRF model lends it to the development of computer techniques which implement automatic error analysis. Since SMIRF contains a full range of simple and complex solutions, the research can develop techniques which work on simple programs first and progress to more complex areas as knowledge grows.

Methodology

This research project was initiated to determine the suitability of using SMIRF as a basis for developing a technique for automatic error analysis within computer models. In order to make such a determination, however, the researcher needed to become much more familiar with the model than was previously the case. Therefore, the researcher and the laboratory

focal point decided that a good general focus for the summer program was to upgrade SMIRF in several areas and to add new capabilities. Time permitting, the researcher would try to obtain a wider distribution release for the code. The following is a list of the goals:

1. Upgrade the SMIRF to calculate the relative aspect angle between the observer and the flare.
2. Add the capability to model a thrusted type of flare.
3. Obtain wider dissemination permission for SMIRF. Currently, SMIRF can only be distributed to the Air Force Information Warfare Center.

Results

The research had mixed success in meeting the original goals. As it happened, the research quickly turned up other requirements which were incorporated into the plan and accomplished. The actual accomplishments are described here.

1. Added the capability for SMIRF to calculate the observer positions as a function of time. This is totally new and leverages work conducted by the researcher's coworkers at Texas Tech University.
2. Added the capability for SMIRF to calculate the relative positions and aspect angles from flare to observer. Previously a separate program.
3. Added the capability for SMIRF to use the observer aspect angle calculations to drive SIRRM. This is totally new and could not be performed until 2. above was performed.
4. Added the capability to run SMIRF as a "trajectory only" code. This eliminated the requirement for a separately compiled code for this purpose.

5. Improved the performance of SMIRF in calculating the trajectory of the flare.
6. Improved the documentation of SMIRF.

This section begins with a discussion of these changes to SMIRF. It concludes with an analysis of the error inherent in the calculation of the flare trajectory.

Observer Position Calculations

The first action scheduled for the summer was not difficult. In fact, it had already been accomplished by another programmer. But the previous capability was limited in several respects.

1. The observer positions were calculated based on the observer traveling in a straight line only.
2. The equation for the line followed by the observer was hard-coded into the program.
3. The program was not integrated into the SMIRF program.

As a result of these limitations, a fresh look at the requirements yielded some new ideas. The code should handle observer positions which are specified independently of the flare positions. It should be integral to the SMIRF program, calculating the observer positions and the range and aspect angles for each output flare position. Finally, this new information should somehow be used to improve the results of the flare signature calculation.

Since the observer positions should be specified independently of the flare calculations, the program would need to interpolate observer positions. As an example of the type of calculation necessary, suppose the observer positions are specified for times beginning $\frac{3}{4}$ second before the flare is dropped and at 1 second intervals. Then if the flare trajectory is calculated for time intervals 0.5 sec apart, the times for the observer positions and for the flare positions do not coincide. Thus there is a need to interpolate the observer position for the instant in time when the flare position is specified.

What type of interpolation should be used? Linear interpolation is easy and can be reasonably accurate if the observer positions are specified with small enough intervals. However, linear interpolation yields a discontinuous derivative in the motion of the observer, not a very realistic result. Polynomial interpolation yields a function which is infinitely differentiable, but can yield unreasonable results when used with the wrong input. An interpolation technique which can yield the desired smoothness, but is relatively tolerant of arbitrary inputs is desired. Piecewise polynomial interpolation can be tuned to the requirements at hand. The desired smoothness provides a nice set of constraints on the system, and the order of the polynomials can be kept low enough to keep the undesirable motion in check. Even though there are some problems with actually trying to fly an aircraft with a cubic spline (or any spline derived from the minimum energy control law [5],[6]), we note that the problems reported by Karlsson are for the extreme of discontinuous input. Thus, we may choose to use the cubic spline for its relative simplicity and accuracy.

The observer positions need to be specified independently from the execution of SMIRF. This is accomplished with a file containing the time and position information about the observer. This file is read at the beginning of a SMIRF run and the data is used to initialize the spline interpolation. Then at each iteration of the flare's trajectory, SMIRF interpolates the location of the observer at that time. Then the next upgrade to SMIRF is possible.

Observer/Flare Relationship and SIRRM Driver

SMIRF computes the range from the observer to the flare and the observation aspect angle by manipulating the spatial positions of the flare and observer as vectors. Since the velocity vector of the flare is known, it is used to create a pointing vector for the flare. Thus these three vectors completely represent the relationship between the flare and the observer. After a series of simple calculations, then SMIRF is poised to output the

information about the relationship.

Previously, the information about the relationship between the flare and observer was written to a file and used by the software user to determine which aspect angles and ranges to use when executing SMIRF. Note that the information about aspect angles was used to guide the execution of SMIRF. Thus if SMIRF could use this information directly in its own execution, the process would be shortened considerably. This is precisely the change which was added to SMIRF.

At each time step, the program copies the information about the current relationship between the flare and observer to a special file. This file is used to tell the Standard Infrared Radiation Model (SIRRM) the altitude of the observer and flare and the range and aspect angle of the observation. Ordinarily this file is created by the user to tell SIRRM this information. In this case SMIRF takes the role of user. SIRRM then calculates the radiation from the flare's plume toward the observer and also computes the change in that radiation due to the atmospheric interference. The use of actual modeled altitudes is vital to the correct calculation of the observed signature. This situation was rarely achieved using the previous method of execution.

"Trajectory Only" Calculation Mode

Previous to this summer, there were two versions of SMIRF: one which calculated only the trajectory of the flare, and another which computed those trajectories and also computed the signature of the flare. There were two versions for the simple fact that computation of the flare signature is very time-consuming. However, the only difference between the two codes was that, at crucial points, the "trajectory only" version didn't call certain subroutines of SMIRF. By combining the two codes, the task of maintenance is greatly simplified.

Trajectory Computation Improvements

The change to how SMIRF computes the trajectory is subtle, but results in a huge saving of computational time in this area. SMIRF computes the trajectory of the flare by solving the differential equation for motion of a sphere through space:

$$\dot{u} = f(t, u) \quad \text{where} \quad f(t, u) = \frac{-C \times u \times |V|}{M_f} + \frac{thrust \times u}{M_f \times |V|} \quad (1)$$

$$\dot{v} = f(t, v) \quad \text{where} \quad f(t, v) = \frac{-C \times v \times |V|}{M_f} + \frac{thrust \times v}{M_f \times |V|} \quad (2)$$

$$\dot{w} = f(t, w) \quad \text{where} \quad f(t, w) = \frac{-C \times w \times |V|}{M_f} + \frac{thrust \times w}{M_f \times |V|} - g \quad (3)$$

And C is a function itself:

$$C = C_d \times A_F \times \rho_\infty \quad (4)$$

with C_d the coefficient of drag for a sphere, A_F the cross sectional area of a sphere equivalent to the flare and ρ_∞ the density of the freestream. These equations were derived for SMIRF by Bauer and Jordan [4]. The computation of the drag coefficient has been the subject of much previous research and currently follows work by Nichols [3].

The differential equation provides information about the velocity of the flare. This is integrated to determine the position of the flare at any point in time. Of interest to us here is the numerical techniques used to solve for the velocity and the position.

The method chosen is a fourth order Adams-Bashforth predictor followed by a fifth order Adams-Moulton corrector. For reasons which will become apparent, this is the best choice for solving equations (1) through (3). The only drawback to the use of an Adams family method is that these methods require multiple steps of solution to solve for the next step. Thus a fourth order Runge Kutta method is used to generate the starting steps for the Adams PECE² method.

²Predict Evaluate Correct Evaluate.

One aspect of the Adams PECE approach which is used to full advantage in SMIRF is the fact that, for each new step in the solutions, only one function evaluation is required. In (4) above, the term for coefficient of drag is definitely not constant. In fact, the computations for this term require many complex operations such as logarithms, exponentials, square roots. The computation of drag coefficient is quite expensive. Thus the Adams PECE method is much more efficient than, say, a Runge Kutta method.

Another aspect of the Adams PECE method has caused it to fall out of favor in recent years. That is the cost of changing the stepsize. If the stepsize is to be reduced, the Adams methods require a complete set of past steps to be created. However, the stepsize never needs to be reduced in our application. In fact, by choosing to only increase the stepsize by doubling, we may change step size with no actual cost. This is accomplished by maintaining a pipeline of past values of the solution. The pipeline is twice the length of that required to support the Adams Bashforth predictor. When the error analysis part of the code determines that error levels have dropped sufficiently³, the differential equation solver sets the step size at double the old size and moves data from the 1, 3, 5, 7 positions into the 0 through 3 positions and the Adams method never notices the difference. Careful examination of this shift will show that the new step size is perfectly represented in the new pipeline (positions 0 through 3).

The result of these changes has been an improvement of more than a factor of two in the computation time for flare trajectory. It should be noted that most of the savings come from the use of variable step sizes.

SMIRF Code Documentation

Prior to this summer, there was no real documentation for SMIRF. The only documentation consisted of a few lines of comments in the FORTRAN code and what could be termed programmers notes⁴. While the interfaces to some of the subroutines were specified and

³see **Error Analysis** section.

⁴Derivation of formulas and such.

some information was provided about the technical aspects of those routines, the bulk of the code was undocumented. The primary reason for this state of affairs was the fact that SMIRF had no permanent project team.

A combination design, theory, and detailed design document was created. It contains that information which existed previously along with updates and additions as necessary to accurately represent the source code as it currently exists[2]. It is hoped that the design document for SMIRF will be updated as changes are made to the program.

Due to the size of the document and the mathematical nature of the design, the SMIRF design document was typeset with L^AT_EX 2_ε.

Error Analysis

Due to the nature of the differential equation being solved, the local truncation error decreases monotonically during the solution process, thus by choosing to increase the step size when the truncation error is small enough, there will be no need to decrease the step size. This provides a very efficient use of the Adams method, since decreasing the step size would require extra function evaluations, whereas in this implementation, when the truncation error estimate falls to a small enough value, we double the step size, set the starting step size equal to the new current step size and proceed.

We wish to estimate and control the global error associated with the numerical solution of the differential equation

$$\dot{v} = f(t, v) \quad \text{with } f(t, v) = -c(v) \times v^2$$

Ordinarily, this might be a relatively easy task, but in this instance it is complicated by the fact that not only does $c(v)$ change during the course of a solution, but the function $f(t, v)$ has derivatives that become increasingly dependent on v and $c(v)$. For example, the solution to initial conditions at $t = 0$, $v = 864$, gives an initial value for c of 4.7. And the conditions after one second of solution, $v = 204$, give a value for c of .67. These values lead to the conclusion that at $t = 0$, f_v , is -8102 . In fact, this value rapidly decreases

to a value of about -560 at $t = 1$. If we were to use the value -8102 for the Lipschitz constant, we would see that many of our estimates for error would be far too conservative. As an added consideration, the value of the fifth derivative is $> 10^{23}$ at $t = 0$ but drops to $< 10^{16}$ at $t = 1$. This implies the conclusion that a value for the step size which works for $t = 0$ may be excessively small for later calculations. As an example, consider a step size of $h = 10^{-5}$. Then since the local truncation error⁵ is given by

$$T_n \approx \frac{251}{6} h^5 c^5 v^6 \quad (5)$$

we have

$$T_n \approx 4 \times 10^{-3} \quad \text{when } t = 0$$

and

$$T_n \approx 4 \times 10^{-11} \quad \text{when } t = 1$$

we can see that the use of a constant value for the step size will result in significant inefficiencies.

There is a way out of this problem. We will use a variable stepsize, but the estimate for global error still depends on c , v , which change drastically during the course of solution.

First, let us reacquaint ourselves with the process we are using and with the analysis for numerical error.

This process depends primarily on the Adams fourth and fifth order predictor corrector pair:

Predictor:

$$y_{n+4} = y_{n+3} + \frac{1}{24}(55f_3 - 59f_2 + 37f_1 - 9f_0) \quad (6)$$

⁵Lambert, p28 and p42

Corrector:

$$y_{n+4} = y_{n+3} + \frac{1}{720}(251f_4 + 646f_3 - 264f_2 + 106f_1 - 19f_0) \quad (7)$$

But this pair is not self-starting, so we choose to use the fourth order Runge Kutta method:

$$Y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (8)$$

with

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$

$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

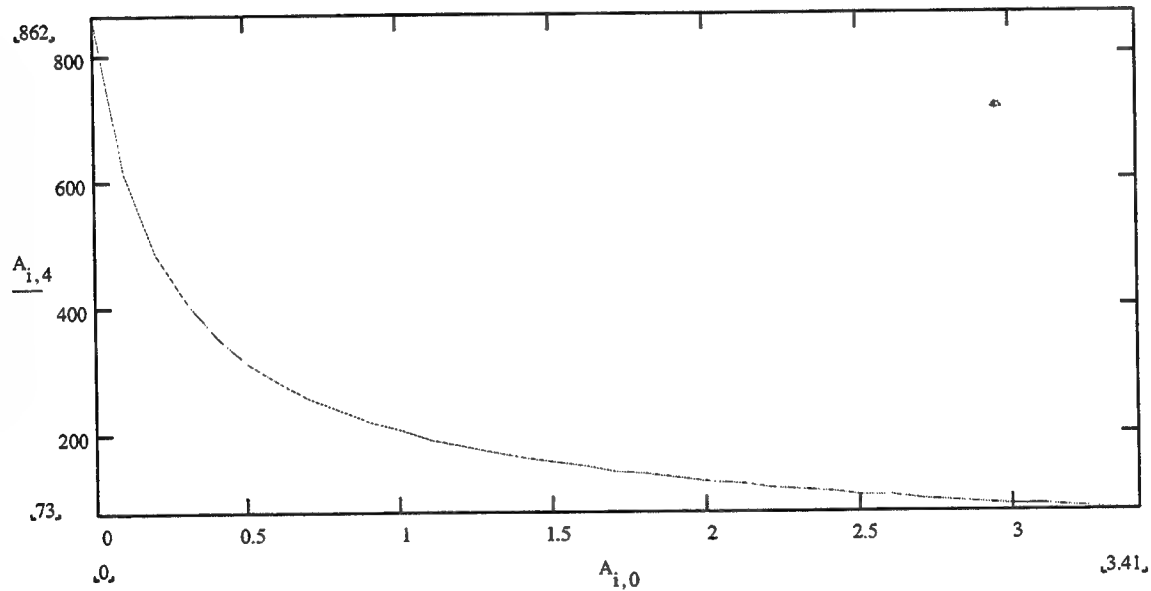


Figure 1: The X velocity of a flare

Ordinarily, we would determine the maximum error on the starting values provided by the RK method, then we would use these to estimate the global error experienced by the Adams PC method. However, since our Lipschitz constant is negative, we know

that the global error will be dominated by the local truncation error. A quick glance at figure 1 shows why this is so. Any error which tends to increase the velocity will result in a corrective tendency in the negative direction, while any error in the negative direction will result in less drag and hence a corrective tendency in the positive direction.

The local truncation is expressed by the formula⁶:

$$Y(x_{n+k}) - y_{n+k} = C_{p+1}h^{p+1}Y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \quad (9)$$

where $Y(x)$ is the "true" solution, y_{n+k} , the "numerical" solution, and⁷ $C_{p+1} = \frac{251}{720}$, which is just (5).

However, since the values usually assumed to be constant decrease so rapidly, this would result in an error estimation which is far too conservative. Therefore, we measure the approximate truncation error during runtime with the Richardson extrapolation approximation:

$$T_n = \frac{(Y_{2h} - Y_h)}{63} \quad (10)$$

Y_{2h} is the AM solution for Y using a step size twice that of Y_h . 63 is $(2^{(p+1)} - 1)$.

Conclusions

The objective of this summer research program was to determine the suitability of the Signature Model for Infrared Flares (SMIRF) as a platform for research into automatic error calculation and estimation within computer programs. In order to enhance this objective, certain upgrades to SMIRF were also accomplished.

SMIRF has had a long career of development and growth. The work accomplished this summer is now part of that history. One positive aspect to the approach taken through

⁶Lambert, p28

⁷Lambert, p42

the years of SMIRF development is that the code has much more capability than would ordinarily be expected of a project with this level of funding. SMIRF is now capable of using data sheet information about a flare, along with flight parameters of both the observer and the flare to compute the observed radiation patterns of the flare. Specifically, SMIRF is capable of calculating the flight path of both the flare and of the observer and determining their relationship at any point in time.

SMIRF has benefitted in other respects also. Among the improvements have been an upgraded user interface and improved maintainability. The former allows the user to specify the observer-flare relationship with greater flexibility. The latter will enable future programmers to add still more upgrades to SMIRF with ease.

In the process of analyzing the trajectory portion of the SMIRF code, certain inefficiencies were identified and removed. This resulted in a quicker more accurate estimation of the flare trajectory. The trajectory code was analyzed for error and the results used in the code for "on-the-fly" algorithm tuning to result in an improved compromise between speed and accuracy.

Finally, SMIRF was analyzed for its suitability as a platform for future research. This analysis indicated that the wide variation in complexity of the models contained in SMIRF makes it ideally suited for this task.

References

- [1] Brown, Kendal K., "A Methodology for Assessing Experimental Uncertainties in Curvefits with Application to Compressor Map Characteristics," Final Report for AFOSR Graduate Student Research Program, Page 1-1, 1995.
- [2] Tomlinson, John R., *The Signature Model for Infrared (IR) Flares (SMIRF)*, Internal Document, AEDC, 1995.
- [3] R.H. Nichols, "Effect of Particle Dynamics on Turbulence Measurements with the Laser Doppler Velocimeter." AEDC-TR-86-41, Dec 1986.
- [4] Memo from R.C. Bauer and J.K. Jordan, Calspan Corporation/ AEDC Operations, May 16, 1990.

- [5] Anders Karlsson, "On the Use of Splines to Construct Aircraft Trajectories," Masters Thesis, 1994, Royal Institute of Technology, Stockholm.
- [6] C. Martin, P. Enqvist, J. Tomlinson, Z. Zhang, "Linear Control Theory, Splines and Interpolation," to appear, *Computation and Control IV*, edited by J. Lund and K Bowers, Burkhauser, 1995.
- [7] Lambert, J.D., *Computational Methods in Ordinary Differential Equations*, Wiley, London, 1973.
- [8] Shampine, L.F., Gordon, M.K., *Computer Solution of Ordinary Differential Equations, The Initial Value Problem*, Freeman, San Francisco, 1975.

Irene De Leon report not available at time of publication.